VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Systém pro odhalování plagiátorství maturitních prací proti internetovým zdrojům

Plagiarism detection system for graduation thesis against internet resources

2012 Marcel Láža

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání diplomové práce

Student: **Bc. Marcel Láža**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 2612T025 Informatika a výpočetní technika

Téma: Systém pro odhalování plagiátorství maturitních prací proti internetovým zdrojům
Plagiarism Detection System for Graduation Thesis Against Internet Resources

Zásady pro vypracování:

Cílem práce je navrhnout a implementovat systém pro odhalování plagiátů u maturitních prací s vyhledáváním shodnosti v internetových zdrojích s využítím stávajícího evidenčního systému pro správu maturitních prací.

1. Definovat pojmy týjkající se odhalování plagiátů.
2. Uvést požadavky stávající legislativy a norem pro korektní uvádění informačních zdrojů ve vědeckých a odborných pracech.
3. Nastudovat a analyzovat metody a nástroje pro odhalování duplicit ve službách internetových protokolů.
4. Vybrat vhodnou metodu, popřípadě navrhnout vlastní pro další využití ve vaší práci.
5. Prostudovat již existující informační systém na správu maturitních prací na vybrané střední škole a definovat jeho možnosti rozšíření k dané problematice.
6. Stanovit důležité podmínky pro zpracování kontrolovaných prací.
7. Na základě zjištěných informací z bodů 1 až 6 navrhnout softwarové řešení pro detekci plagiátů rozšiřující již existující systém pro evidenci maturitních prací.
8. Implementovat navržený systém a uvést jej do provozu.
9. Zhodnotit výsledný softwarový produkt a splnění cílů.

Seznam doporučené odborné literatury:

Podle pokynů vedoucího diplomové práce.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Zdeněk Biolek, Ph.D.**

Konzultant diplomové práce: doc. Ing. Michal Krátký, Ph.D.

Datum zadání: 19.11.2010

Datum odevzdání: 04.05.2012

VYSOKÁ ŠKOLA BÁŇSKÁ-TECHNICKÁ UNIVERZITA Fakulta elektrotechniky a informatiky OSTRAVA

doc. Dr. Ing. Eduard Sojka
vedoucí katedry

prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 4. května 2012

Marcel Fořt .....

Rád bych na tomto místě poděkoval panu Ing. Zdeňkovi Biolkovi, Ph.D. za vedení diplomové práce a panu Ing. Lukáši Haplovi za konzultace, které byly nezbytné pro zjištění vstupů k vytvoření požadovaného systému, protože bez nich by tato práce nevznikla.

Abstrakt

Cílem diplomové práce je navrhnout a implementovat pomůcku vhodnou pro odhalování plagiátů maturitních prací, která umožní vyučujícím získat důležité informace o jednotlivých prací s možností vyhledávat shody v internetových zdrojích. Dále obeznamuji čtenáře se stávající legislativou a platnými normami, které definují pojem plagiát. Zohledňuji jejich použití na chování středoškolských studentů při tvorbě děl a využívám tyto informace pro návrh systému s ohledem na technické možnosti a vybavení školy. A současně interpretuji existující principy a postupy, které se používají pro detekci plagiátů. Následně ze získaných informací a požadavků vznikne aplikace, která bude využívat vhodný způsob pro odhalování plagiátorství. Tímto také stanoví důležité podmínky a postupy pro zpracování prací se zaměřením na dílčí části. Posledním bodem bude uvést systém do provozu a získat zpětnou vazbu na další úpravy.

Klíčová slova: autorský zákon, plagiát, detekce, dolování textu, slovník, normalizace, SQL, Java, kopie, citace, nelegální, analýzy, API

Abstract

The aim of the thesis is to design and implement a suitable tool for detecting plagiarism of leaving work that will enable teachers to obtain important information about individual works with the option to search for consensus on Internet resources. Familiarize the reader with current legislation and applicable standards that define the concept of plagiarism. Consider their use on the behavior of high school students in the creation of works and use this information to design the system with respect to the rechnical capabilities and qeuipment of the school. At the same time interpret the existing principles and procedures that are used to detect plagiarism. And them from the collected information and requirements create an application that will use the appropriate method for detecting plagiarism. This also provides important conditions and procedures for processing their work, focusing on individual parts. The last point is to setup and start the system and obtain feedback for futhr editing.

Keywords: copyright Act, plagiarism, detection, text mining, dictionary, normalization, SQL, Java, duplicate, citation, illegal, analysis, API

Seznam použitých zkratek a symbolů

IS	–	Informační systém
API	–	Aplication programing interface - rozhraní pro programovací jazyky
RAKE	–	Rapid automatic keyword extraction - rychlá automatická extrakce klíčových slov
SQL	–	Structured Query Language - dotazovací jazyk sloužící k získání dat z databáze
TCP/IP	–	Transimition Control Protocol/Internet Protocol - komunikační protokol sloužící k přenosu dat mezi aplikacemi
JRE	–	Java Runtime Enviroment - Java prostředí určené pro spouštění aplikací
JDK	–	Java Development Kit - Java prostředí určené pro vývoj aplikací, obsahuje rozvněž JRE
XML	–	eXtensible Markup Language - rozšiřitelný značkovací jazyk
PATH	–	proměnné prostředí operačního systému
YQL	–	Yahoo! query language - vyhledávací služba určená developerům využívajících platformy Yahoo!
HTML	–	HyperText Markup Language - značkovací jazyk pro hypertext, využíván při tvorbě Wolrd Wide Webu
PDF	–	Portable Document Format - přenosný formát dokumentů
JDBC	–	Java Database Connectivity - je API, které umožňuje přistupovat k relačním databázím v programovacím jazyce Java
SVD	–	Singular value decomposition - matematická metoda používaná pro extrakci sémantických vztaů

Obsah

Seznam tabulek

Seznam obrázků

Seznam výpisů zdrojového kódu

1 ÚVOD

Účelem této diplomové práce je osvětlit problematiku vzniku plagiátorství v souvislosti s platnou legislativou, zmínit jeho negativní dopad a také objasnit příčinu vzniku této problematiky a navrhnout jak jej řešit a předcházet mu. Dále prostudovat existující metody a definované postupy pro zpracování a analýzu textu, které umožňují úspěšnou detekci plagiátů.

Nastudování důležitých materiálů povede k vytvoření softwarové aplikace, která bude využívat vyhledávacích systémů třetích stran k odhalování plagiátů pomocí dokumentů dostupných na internetu. Vyhledávacími systémy je myšleno využití služeb poskytovanými společnostmi jako Google, Bing, Yahoo apod. s ohledem na jejich cenu. Systém bude moci pro začátek umožňovat zpracování prací ve formátu PDF a s předem nadefinovanou šablonou. Na základě definovaných požadavků není třeba, aby systém nabízel grafické rozhraní. Dle zadaných případů užití postačí pouze nastavení potřebných parametrů jako přístup ke zpracovávaným souborům, databázi a internetu. Součástí bude také vytvoření malé webové aplikace, která umožní zobrazit zpracované výsledky. Účelem je oddělit tyto dvě části a neumožnit vzájemnou interakci z důvodu, že se práce budou zpracovávat až po termínu odevzdání a nemá sloužit k tomu, aby případný plagiátor měl možnost práci opravit. Smyslem produktu není, aby určoval, kterého hříšníka potrestat, ale aby poukázal na možnosti vzniku plagiátů u středoškolských studentů a aby díky němu vznikla zpětná vazba a docházelo k využití těchto informací pro prevenci vzniku plagiátů (například vyučující by měl tak možnost zjistit příčinu vzniku plagiátu a podle toho jednat).

Jedná se tedy o systém, který bude sloužit pro účely pedagogických pracovníku, kteří budou schopni jednoduše zjistit, zda-li odevzdaná maturitní práce je plagiátem. Samozřejmostí je poskytnout veškerou dokumentaci jak uživatelskou tak programátorskou, aby v případě osvědčení metod mohlo dojít k jeho rozšiřování.

2 PLAGIÁTORSTVÍ

2.1 Definice pojmu plagiátorství

Nejdříve si řekneme, co pojem plagiátorství znamená. Mnoho lidí se domnívá, že jde o kopírování případně krádež originálního nápadu, který vymyslel někdo jiný. Zjednodušeně by se dalo odpovědět ano, ale je rozdíl krást a vydávat myšlenky jiného autora za vlastní nebo vycházet z nich a uvést originální nápad případně stávající zdroje. Jinak řečeno jedné se o podvodný čin, který by se dal posuzovat až jako trestný čin. Plagiátorství je problém, který zasahuje nejen do školského prostředí, ale prakticky do celé společnosti.

Vezměme v potaz modelovou situaci např. firma vyrábějící televizní vysílače a přijímače vloží obrovské množství peněz do výzkumu technologií, aby umožnila přijímat několik různých typů vysílání (vysílání v běžné kvalitě až po HD kvalitu). Dodá tedy na trh univerzální produkt, který umožňuje zpracovávat mnoho různých formátů a díky tomu se z něj stane prodejní trhák. Problém nastane v okamžiku, kdy jiný výrobce využije úspěch původního výrobce pro sebe. Výhodou je, že nemusí vynakládat tak velké množství peněz do vývoje, použije mnoho idejí z původního výrobku, provede kosmetické úpravy a prodává jej za vlastní. Obě zařízení jsou obdobná, nabízejí stejné funkce a to, které vychází z původního je s největší pravděpodobností levnější. Tím parazituje původního výrobce, který vynaložilo mnoho financí do originálního produktu a připravuje jej o zisk.

V tomto případě se jednalo o duševní vlastnictví, které má i hmotnou podobu, takže je jednodušší dokázat, že se opravdu jedná o plagiát.

Jiným příkladem je hudební průmysl, kde posluchač zaslechne píseň, která mu zní povědomě a pomyslí si, že už tuto písničku někdy slyšel. Pokud se tak děje pravděpodobně zaslechl píseň od plagiátora. Někdo by mohl namítat, že jakékoliv texty, noty, znělky smyčky apod. jsou v hudebním průmyslu licencované. To jsou, ale pokud někdo pozmění jednotlivé části, změní v některých místech tóninu apod., tak už je to mnohem hůře napadnutelné.

Dostáváme se tedy do místa, jak správně definovat, kdy se jedná o plagiát a kdy nikoliv. Nemůžeme považovat za plagiát to, co vychází z všeobecně známého faktu, to se týká především duševního vlastnictví [16].

Můžeme respektive musíme vzít v úvahu co definuje zákon 121/2000 Sb. Zákon o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) hlava 1, právo autorské, Díl 2 Autorství paragraf 5 Autor, který zní: *„Autorem je fyzická osoba, která dílo vytvořila. Autorem díla souborného jako celku je fyzická osoba, která je tvůrčím způsobem vybrala nebo uspořádala tím nejsou dotčena práva autorů děl do souboru zařazených“* [15].

Naneštěstí autorský ani jiný zákon nedefinuje pojem plagiátorství. Ovšem to je uvedeno normlou ČSN ISO 5127-2003, která definuje plagiát jako *„představení duševního díla jiného autora půjčeného nebo napodobeného v celku nebo z části, jako svého vlastního“* [16]. To znamená, pokud někdo použije jakoukoliv myšlenku v odborné práci, bez toho aniž by uvedl původního autora, dopouští se plagiátorství. Pokud ovšem uvedu autora původní

myšlenky, s ohledem na normy pro korektní uvádění citací díla, tak už se plagiátorství nedopouštím.

2.2 Příčina vzniku plagiátorství

Nevýhodou dnešní doby je, že díky informačním zdrojům jako jsou odborné články, knihy, publikace, školy a internet, máme přístup k myšlenkám a nápadům jiných prakticky z jakéhokoliv místa. Všechny tyto informace vnímáme, vstřebáváme a nespočetněkrát využíváme ve vlastních pracích. Tím, že se lidstvo vyvíjí raketovým tempem umožnilo právě sdílení informací.

Podíváme-li se do historie zjistíme, že některé vynálezy, nápady a myšlenky vznikly na několika místech zároveň např. bleskosvod Benjamina Frenklina, který nepracoval úplně ideálně a bleskosvod Prokopa Diviše, který byl navíc uzemněn a oproti Frenklinovu, fungoval dokonale. Podle literárních pramené, které existují není uplně jasné, kdo z pánů „opisoval" a kdo z nich vycházel z jakých pramenů. Dalším příkladem může být vynález dalekohledu, který je přisuzován italskému vědci Galileo Galilei, jenže první patent na vynález dalekohůedu je přisuzován holandskému optikovi Hansu Lippersheymu [28] [29]. Sdílení informací samozřejmě existovalo i tehdy, ale nebylo tak snadné jak dnes, kdy se lidé nemohli tak snadno dovědět, kdo pracuje na konkrétních věcech. A také nebylo tak jednoduché právě kvůli sdílení informací využívat schopnosti jiných a spolupracovat ve větším měřítku než je možné dnes, každý vědec tak sám za sebe musel nastudovat obrovské množství informací. Dnes především z pohledu studenta, díky čím dál rychlejšímu sdílení informací, kdy je možné dovědět se děj na opačné straně polokoule behem pár vteřin, je zbytečné učit se to co už někdo před ním vymyslel. Proto z velké většiny případů dochází k vzniku plagiátu především na akademické půdě. Chybí zde základní motivace něco sám za sebe dokázat, protože student nemá chuť vymýšlet už dávno vymyšlené. Bohužel si neuvědomuje souvislosti mezi studijní náplní a to, že pokud bude mít alespoň základní přehled z širokého spektra vědomostí, bude pro něj mnohem snažší učit se a chápat mnohem složitější látku a vypracovávat se dál. Člověk je od přírody líný a všechny nápady měly společný účel a to usnadnit si život, jenže s příchodem nových a nových informací už nejsme schopni je ani všechny pojmout a dokonce si ani vybrat, kterým směrem se budeme ubírat.

Ovšem musíme nahlížet na problém vzniku z několika úhlů, protože jsou obory a odvětví, kde není tak jednoduché, aby vznikl plagiát. Jedná se o rozdíl mezi formální (filosofie, matematika, logika) a reálnou vědou (přírodní, humanitní), technické, lékařské nebo také o ty, které jsou exaktní a které nejsou.

Příčinou je tedy informovanost až přesycenost informacemi. Především s dostupností počítačových síti, internetu, elektronickým archivům, knihovním databázím, do kterých může nahlédnout kdokoliv a vytáhnout si všechny dostupné informace, i proto dochází ke vzniku plagiátorství v mnohem větší míře. Dříve také šlo vytvořit plagiát, ale bylo k tomu potřeba mnohem větší úsilí než dnes [20] [26].

2.3 Řešení a prevence proti vzniku plagiátů

Řešení problému plagiátorství není jednoduchý úkol a určitě ani není jednoznačný, vezmeme-li v potaz autorský zákon a i to, co uvádí normy. Nelze úplně tedy definovat, jak plagiáty odhalit a jak tento problém řešit. Musíme si tedy uvědomit čeho chceme docílit a dle toho stanovit prostředky, jak tento problém řešit. Např. z pohledu školství nechceme, aby se studenti dopouštěli plagiátorství, musíme je tedy upozornit a informovat o nebezpečí a možných sankcích, jakých by se v případě vzniku plagiátu dopouštěli. Jedním slovem jedná se o prevenci [20].

Preventivním řešením je tedy vést studenty od prvních kroků k jejich vlastní tvorbě. Jakožto osoba vzdělávající at'už je to vyučující, mentor, profesor jsme si schopni uvědomit situace, kdy se student může dopustit plagiátorství a měli bychom jej na tuto situaci upozornit a případně mu pomoci, jak se tomuto problému vyhnout. Tzn. že by se studenti měli naučit pracovat s cizími myšlenkami, protože není možné, aby student používal výhradně své. Například lze jejich práce at'už semestrální, maturitní a jiné směrovat, tím že zadáme seznam zdrojů, který budou studenti muset použít, tím se můžeme vyhnout problematice plagiátorství, protože student bude předpokládat, že máme přehled, co se v daném zdroji nachází a nebude jej moci tak jednoduše zkopírovat.

Norma, která definuje plagiát říká, že k plagiátorství dochází i v případě, že dojde k vypuštění vět nebo jednotlivých slov, nebo nahrazení slov synonymy a nebo také záměně pořadí slov v případě, že neuvedeme autora původní myšlenky [16]. Máme k dispozici tři způsoby jak uvést autora původní myšlenky:

1. **Nepřímá citace** (parafráze původního textu) - postup, kdy využijeme stejnou myšlenku a vyjádříme stejný obsah jiným způsobem tj. formulujeme jej vlastními slovy. v tomto případě jsme povinni uvést zdroj, ze kterého jsme čerpali

2. **Parafráze s doslovnými citacemi** - obdobná situace, kdy využijeme stejnou myšlenku a vyjádříme stejný obsah jinými slovy, ale použijeme část textu, která pochází z originálu. Ten vždy musíme uvést v uvozovkách a samozřejmě také zmínit odkud jsme čerpali. Je vhodné jej vizuálně odlišit například použitím kurzívy.

3. **Přímá citace** - postup, kdy text v původním znění zkopírujeme a bez jakékoliv změny použijeme v textu. Takový text vložíme opět do uvozovek a za uzavírací uvozovkou umístíme číselný odkaz na původní zdroj informace, který většinou uvádíme na konci celé práce v seznamu použité literatury. Opět je vhodné využít kurzívu pro odlišení textu, v případě rozsáhlejšího rázu použít samostatný odstavec.

Posledním řešením, které je silnějšího až brutálního rázu je zavádění informačních systémů, které jsou určeny pro odhalování plagiátů. Silnějšího znamená, že dle použitých algoritmů a nastavené daného systému můžeme odhalit různé úrovně plagiátorství. Brutálním je myšleno to, jak se bude s případným nálezem nakládat a co může plagiátor očekávat za sankce. Jestli pouze pokárání, snížení známky, nutnost přepracování práce až po případné vyloučení. Při nastavení vhodných kritérií se dá takový systém použít jako prevence. Pokud odhalíme plagiát autora, přimějeme ho, aby práci přepracoval a díky

zpětné vazbě, kterou nám systém poskytl se zamyslíme, v čem jsme mohli při výkladu udělat chybu, že jsme například nezaujali všechny posluchače a přehodnotíme náš postoj a obsah výuky.

Rozdíl mezi výukou na střední škole a na vysoké škole je v přístupu pedagoga ke studentům. Účast na vzdělávání v obou institucích je sice dobrovolná, ale v případě střední školy, musí učitelé přistupovat k výuce jednotlivých předmětůtak, že nepředávají pouze konkrétní informaci, ale také pomáhají studentům, jak s danou informací naložit. Takový přístup na vysoké škole ani není možný vzhledem k počtu studentů, pedagogů a studjních oborů. Střední škola je odlišný institut, který předává základní znalosti a informace do života a vysoká škola umožňuje takto získaný základ rozšiřovat. Proto se na středních školách přistupuje častěji k metodám prevence a poskytování dalších šancí, kdy student může svůj prohřešek odčinit.

2.4 Systémy pro detekci plagiátů

V dnešní době rychle se rozrůstajících technologií a rychlosti sdílení je poněkud nemožné obejít se bez systémů na odhalování plagiátů, proto vznikají. Základním principem, který byl použit v ranném vývoji všech těchto nástrojů, postupů a metodik, bylo nesporně testování shod mezi textovými dokumenty, kdy se porovnával obsah a části textu na identickou shodu. Museli jsme tedy disponovat takovými sobory, které jsme mohli porovnávat. To postupně vedlo k vytvoření centrálních repozitářů, které nám umožnily si evidovat a rozšiřovat tak databázi dokumentů a informací pro budoucí porovnávání prací, které s postupem času přibývají [21].

Takovým příkladem systému, který využívá centrálního repozitáře je projekt Theses.cz, který byl vyvinut na Fakultě informatiky Masarykově Univerzitě v Brně, kdy roku 2008 byl vznik projektu finančně podpořen z Centralizovaného projektu MŠMT C1/2008 *„Národní registr VŠKP a systém na odhalování plagiátů“* a v roce 2011 byl opět systém finančně podpořen z Centralizovaného rozvojového projektu MŠMT C39/2011 *„Meziuniverzitní síť technických a metodických opatření na ochranu proti plagiátorství “*, díky čemuž jej mohou využívat univerzity v celé České republice [21].

Z hlediska centrálního repozitáře se předpokládá neustálá expanze, kdy s časem budou přibývat další práce nejen z jedné univerzity, ale z více škol. Výhodou takového typu archivu je, že lze nastavit přístup k repozitáři buď jako soukromý nebo jako veřejný. k pracem označeným jako soukromé budou mít přístup pouze oprávnění uživatelé například vyučující, vedoucí apod. k veřejně označeným pracím bude umožněn přístup všem, kteří si budou moci prohlédnout danou práci a podobnost s jinými, případně jim bude umožněno porovnat takovou práci se svým vstupem. Nastavení viditelnosti pro veřejný sektor nejenom pro studenty může být také z části nevýhodou, protože tím umožníme vznik dalších plagiátů.

Zabstraktníme-li myšlenku centrálního repozitáře můžeme využít samotný Internet jako centrální repozitář elektronických dokumentů. Když máme k dispozici vyhledávací nástroje jako např. Google, Bing, Yahoo, WhatUSeek a dalších, které mají k dispozici i programová rozhraní nebo-li API, neměl by zde být z technického hlediska problém vyhledávat a porovnávat obsah samotných dokumentů na internetu. Musíme se ovšem

zamyslet, jestli tak obrovská databáze může být přínosem. Ano, obsahuje nepřeberné množství informací nejen ze školského sektoru, ale i z jiných institucí, které zveřejňují své aktivity, výsledky a práce. Určitě to má velký potenciál především z pohledu ostatních autorů, kteří nepůsobí na školách, ale pracují např. v komerční sféře, ti také nechtějí, aby jejich práce byly zneužívány a kopírovány. Na první pohled se to zdá být velkým přínosem, problémem ovšem může být to nepřeberné množství informací a jejich nestálost. Internet je z hlediska takového repozitáře nepoužitelný pro neustálé porovnávání, je nestabilní, protože informace, které nalezneme dnes, se nemusejí na stejném místě vyskytovat zítra. Toto nám nikdo nezaručí, takže ho můžeme použít pouze jako prostředek pro vyhledání dokumentů dle zadaných klíčových slov, ty si poté uložit zanalyzovat a porovnat.

2.4.1 Existující systémy pro odhalování plagiátů

S problémem vzniku plagiátů se nepotýkáme pouze u nás, jedná se o globální problém a nejsme jediní a ani první, kdo se jej snaží řešit a bojovat proti němu všemi dostupnými prostředky. Pokud byste se chtěli podívat jaké systémy se používají v zahraničí, tak je můžete vyhledat pod pojmem „detection tools“. Těchto systému existuje poměrně hodně, ale většina z nich je zpoplatněna. Není se čemu divit, jedná se o byznys a žádné zpracování velkého množství textu a porovnávání nemůže být zadarmo. Systémy se dají rozdělit do několika kategorií, podle toho jak nakládají s detekcí plagiátů. Existují tři hlavní případy, podle kterých mohou být dokumenty kontrolovány. Některé specifické případy budou spadat do dvou nebo i do všech tří kategorií. Musíme si tedy uvědomit rozdíly mezi těmito případy, abychom mohli dále mohli vybírat vhodné nástroje pro detekci kopií. Je to stejné jako bychom povolovali matici kladivem. Také musíme vybrat vhodný nástroj [22].

1. **Ověření originality** - představte si situaci, se kterou se určitě setkala většina učitelů, profesorů a dalších pedagogických pracovníků, kdy máte porovnat obsah článku, textu, seminární práce apod. A je třeba zjistit jestli je práce původní nebo ne. v tuto chvíli je pro nás nejdůležitější obsah a přesnost databáze. Účelem není vyhledat každý výskyt podobnosti, ale jednu konkrétní práci, která by mohla být původním zdrojem.

2. **Sledování obsahu zneužití** - tato část je dnes nejvíce využívaná pro detekci plagiátů. Například situace, kdy kontrolujete autentický článek, u kterého víte pravý původ a chcete zjistit, jak se informace, které osahuje šíří a využívají v jiných dokumentech například na webu. Nezajímá vás pouze jeden výskyt, ale všechny, které jsou k dispozici. Stále je důležitý obsah a přesnost databáze, ale o něco méně než v předchozím případě. Důvodem je větší tolerance falešných poplachů, protože je zde zjednodušeno porovnávání z toho hlediska, že jste schopni si vizuelně porovnat všechny výskyty. Nejedná se zde o porovnávání prací o desítkách stran, ale pouze malého článku, odstavce apod. Důležité je vrátit velké a pokud možno co nejvíce přesné množství výsledků.

3. **Hloubková analýza** - poslední situace je taková, že jste k porovnávanému dokumentu našli několik kandidátů, které by mohly být zdrojovými, ideálně pokud se dostanete k jednomu konkrétnímu dokumentu. Kontrola musí tedy potvrdit, zda-li se jedná o plagiát nebo z jak velké části se jedná o plagiát. Pokud tedy máme oba dokumenty k dispozici tak už se nezaměřujeme na další hledání, ale musíme využít pokročilých technik analýzy textu a porovnat oba dokumenty do hloubky a získat detailní výsledky.

Ve výsledku pro hledání shodností na v rámci internetu budeme potřebovat využít kombinaci všech tří metod. Za prvé vyhledání, výskyt at' už více nebo pouze jednoho a následné podrobné porovnání, ovšem ne na úrovni celé práce, ale jednotlivých částí [22].

Samozřejmě je zde více odlišností mezi jednotlivými nástroji, ale z komplexního pohledu se tyto tři jeví jako nejvíce používané. Každou metodu lze zdokonalit a časem se ukáže, kam se detekce plagiátů bude vyvíjet.

Aplikace těchto typů existují ve dvou verzích. Prvním je „krabicový typ“, které jsou dostupné ke stažení a jejich funkce je umožněna podle typu použití. Závisí tedy na verzích těchto produktů a také na ceně, jestli je pro studenty, vyučující či celé instituce a podniky. Výhodou takových řešení je, že si vytváříte vlastní repozitář porovnaných dokumentů. Druhým typem jsou aplikace online dostupné přímo přes webový prohlížeč. Technologický boom s příchodem Web 2.0 a HTML 5 a mnoha dalších umožnil přenesení vývoje aplikací z desktopových provedení na web. Má to nesporné množství výhod a samozřejmě nevýhod. z pohledu detekce plagiátů je tento způsob velmi přínosný, protože zveřejněním tohoto systému jako služby se nám repozitář prací může rychle rozrůst. Vždycky ale záleží v jak velké šíři jej zpřístupníme, protože se nám systém může začít plnit zbytečnými a nesmyslnými dokumenty, ale to už je na konkrétní definici podmínek pro využívání. Většinou tyto typy systémů fungují po registraci jednorázově zdarma s tím, že výsledky včetně dokumentu pro porovnání jsou po několika dnech automaticky vymazány.

Následující čtyři systémy dříve patřily do jedné skupiny, která v sobě spojovala funkcionalitu pro detekci plagiátů známá pod názvem Turnitin(WriteCheck) dnes vystupující pod značkou iParadigms. S rostoucím objemem porovnávaných dat bylo zjištěno, že je potřeba rozdělit systém a specializovat se na konkrétní oblasti. Tabulka 1 a tabulka 2 zobrazuje seznam dostupných služeb, které slouží k odhalování plagiátorství.

2.4.2 Nástroj Theses.cz

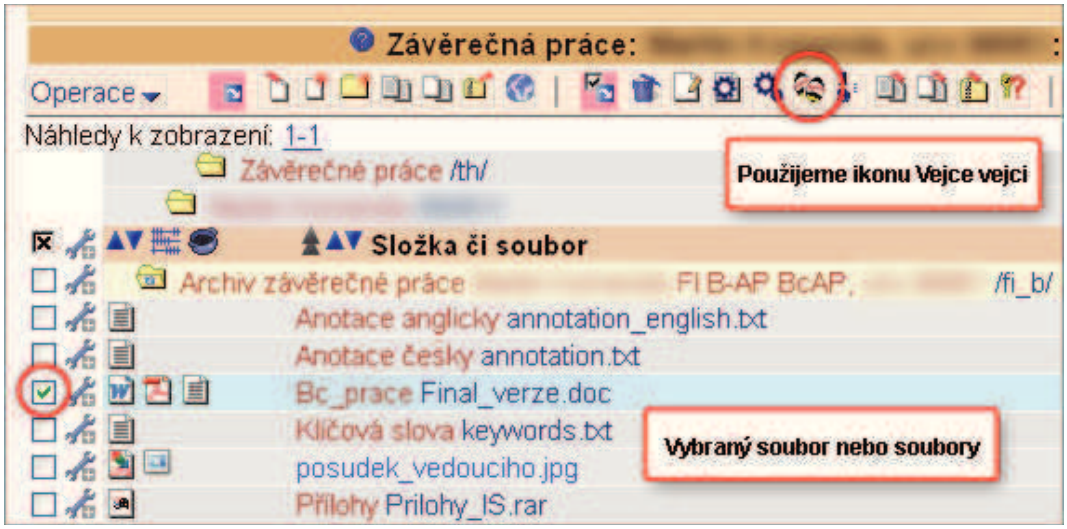
Theses.cz je nejznámější softwarový nástroj pro odhalování plagiátorství v České republice. Byl vyvinut a je provozován Masarykovou univerzitou v Brně a od roku 2008 plní funkci národního registru závěrečných prací. Původ tohoto projektu je datován k roku 2004, kdy do již existujícího informačního systému (IS) byl zabudován registr závěrečných prací. v té době sloužil pro odevzdávání prací v elektronické podobě, díky čemuž vznikl elektronický archiv. Všichni uživatelé IS měli přístup ke všem pracím, které obsahoval. Později vznikl e-learningový systém pro elektronickou výuku a pedagogičtí pracovníci vznesli požadavek, aby jejich práce, články a publikace určené ke studijním účelům ne-

Název	Popis	Cena
Turnition for Educators	Systém určený pro vyučující pro prevenci v boji proti vznikajícím plagiátům. Slouží také jako centrální systém pro odevzdávání prací. http://www.turnitin.com/	Závisí na velikosti univerzity, $1000 - $10000 za rok
WriteCheck	Určeno pro studenty pro kontrolu jejich prací před tím, než ji budou odevzdávat. Využívá stejnou technologii pro porovnávání jako předchozí systém. Pro studentské použití je jednoduchý a navíc nabízí správnou kontrolu citací původních zdrojů, kontrolu gramatiky, její použití a styl písemné práce. https://www.writecheck.com/static/home.html	$7
iThenticate	Používán profesionálními vydavateli, aby mohli zkontrolovat své materiály předtím, než je někde publikují. Tím, že se neustále rozšiřuje objem informací na webu tak by vydavatelé měli zajistit tento typ kontroly. Používají je výzkumná zařízení, právní a finanční instituty, autoři a dokonce i vládní agentury. http://www.ithenticate.com/	$50 za rukopis až po roční předplatné dané smluvními podmínkami
Safeassign	Služba určena pro pedagogické pracovníky k zajištění prevence vzniku plagiátů ve studentských pracích. Technologicky je založena na starší službě MyDropBox, která už není k dispozici a nabízí větší stabilitu, vykon a integraci s ostatními produkty Blackboardu. http://safeassign.com/	Součástí služby Blackboard po jejímž zakoupení je ihned použitelná
Copyscape	Vyhledávaní podobnosti webových stránek. Po zadání odkazu na hlavní stránku započne zpracování obsahu stránky a poté vyhledávání obsahu na webu. http://copyscape.com/	5centů za jedno hledání
Plagiarism Detector	Další nástroj pro odhalování copy and paste dokumentů. Využívá vyhledávacích služeb Bing, Google a Altavista. Podporuje mnoho textových formátů (PDF, DOC, RTF, HTML, PPT) a lze jej importovat do textových editorů. Výsledek porovnávání je velmi přehledný a je zobrazen pomocí grafů s barevným rozlišením shod. http://plagiarism-detector.com/	Demo zdarma. $49 - $99
EVE2	Essay Verification Engine, dle zpracovávaného obsahu najde webové stránky, kde by se daný text mohl nalézat a poté porovnává testovaný soubor s obsahem, který se ve vyhledaných výsledcích nachází. Výsledný report poté zobrazuje procentuální podobnost textu se stránkami. http://www.canexus.com/	$30 neomezené použití

Tabulka 1: Přehled zpoplatněných systémů určených pro odhalování plagiátorství

Název	Popis	Cena
Plagium	Jedná se o systém zdarma, který porovnává obsah na internetu s vloženým textem, který má omezení na 25000 znaků. Využívá API Google a Bing vyhledavace. Umožňuje vyhledávat v 6 různých jazycích. http://plagium.com/	Zdarma, je možné jej sponzorovat.
Jplag	Nástroj pro odhalování plagiátů v zdrojových kódech. Teoreticky jej lze použít i pro porovnávání textových souborů. https://www.ipd.uni-karlsruhe.de/jplag/	Zdarma, nutná registrace
Plagiarism Checker	Umožňuje zpracovat několikastránkové dokumenty a vyhledat je na webu. Odhaluje shodné text tj. metoda copy and paste. http://www.dustball.com/cs/plagiarism.checker/	Zdarma
Moss	Jak už z názvu vypovídá „Meassure of software similarity“, také porovnává obsah zdrojových kódů podobně jako Jplag. http://theory.stanford.edu/ aiken/moss/	Určen pouze pro pedagogické pracovníky Stanford Univerzity.
Viper(Scan My Essay)	Nabízí možnost testování na souborech uložených na lokálním PC a také proti obsahu publikovaném na internetu. Dokument má neomezenou velikost a je porovnáván proti 10 biliónům zdrojům. Na stránkách produktu je porovnáván se systémem turnitin a je údajně lepší a také je zdarma. http://www.scanmyessay.com/	Zdarma

Tabulka 2: Přehled systémů určených pro odhalování plagiátorství zdarma



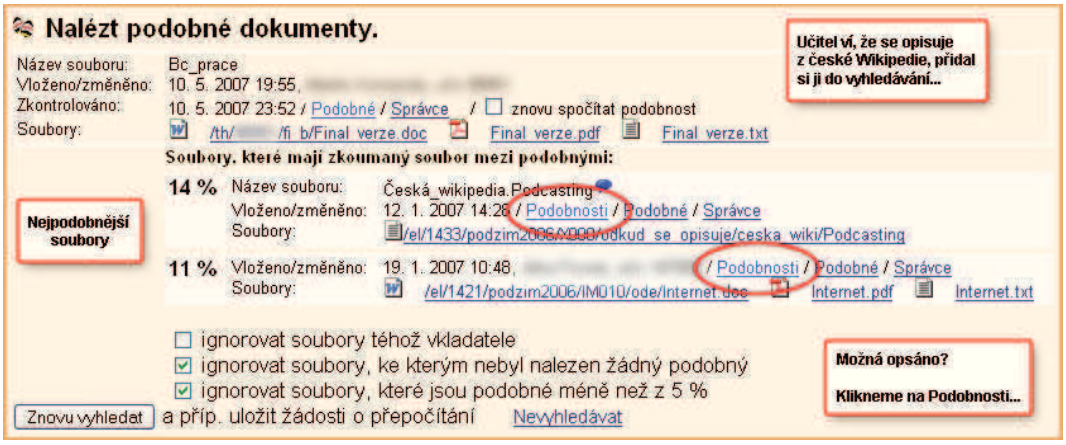
Obrázek 1: Učitel označí dokument(y) a klikne na ikonu Vejce vejci

bylo jednoduché zneužít. v polovině roku 2006 byl IS obohacen o nástroj k porovnávání plagiátů. Na konferenci Inforum 2007 vzbudil obrovský ohlas a získal ocenění, navíc akademická obec rozhodla, že bude vytvořen celonárodní projekt. Při spolupráci s VŠE v Praze ke konci roku 2007 získal projekt podporu ministerstva školství a vznikl nový centralizovaný rozvojový projekt MŠMT C1/2008 s názvem „Národní registr VŠKP a systém na odhalování plagiátů“, který je také financován ministerstvem školství. Vznikl tedy systém využívající principu centrálního repozitáře, obsahující všechny odevzdané bakalářské, diplomové, disertační a rigorózní práce. Kterýkoliv pedagogický pracovník může v případě, že má podezření ohledně pravosti některé z odevzdaných prací, využít možnosti ji přes webové rozhraní IS otestovat. Danou práci nahraje do systému a výsledkem porovnávání je seznam orací, které jsou více, či méně podobné vstupnímu dokumentu. *„Vložený soubor je analyzován a zpracován tak, že je připraven pro vyhledání podobností. To nastává ve finální fázi, kdy uživatelé použijí jednu z funkcí pro vyhledání potenciálních plagiátů. Díky více fázím je vyhledání rychlé a proběhne během několika sekund“* [23].

Uživatelům se nabízejí dvě možnosti jak funkci vyhledávání použít. První se nabízí funkce „jako vejce vejci“ viz Obrázek 1, která vyhledává plagiáty typu „copy/paste“ a druhá, která umožňuje vyhledávat plagiáty z globálního hlediska a vypisuje všechny nalezené podobnosti.

Sytém po provedení hledání, uživateli nabídne u všech nalezených souborů, prohlédnout si nalezené výsledky a jejich podobnost s porovnávaným textem viz Obrázek 2. v případě, že výsledkem hledání jsou větší části textu, které nejsou ocitovány podle platných norem, porovnávaný soubor je s největší pravděpodobností o plagiát. Na základě zobrazených výsledků viz Obrázek 3 učitel se rozhodne, zda-li se jedná o plagiát nebo ne.

Samozřejmě systém nevyužívá pouze výše zmíněného způsobu odhalování, ale počítá i s tím, že se autor plagiátu snažil pozměnit originální text, aby nebylo tak snadné jej odhalit. Tzn. že systém umí porovnat i změnu slovosledu, vynechání slov, jejich záměnu

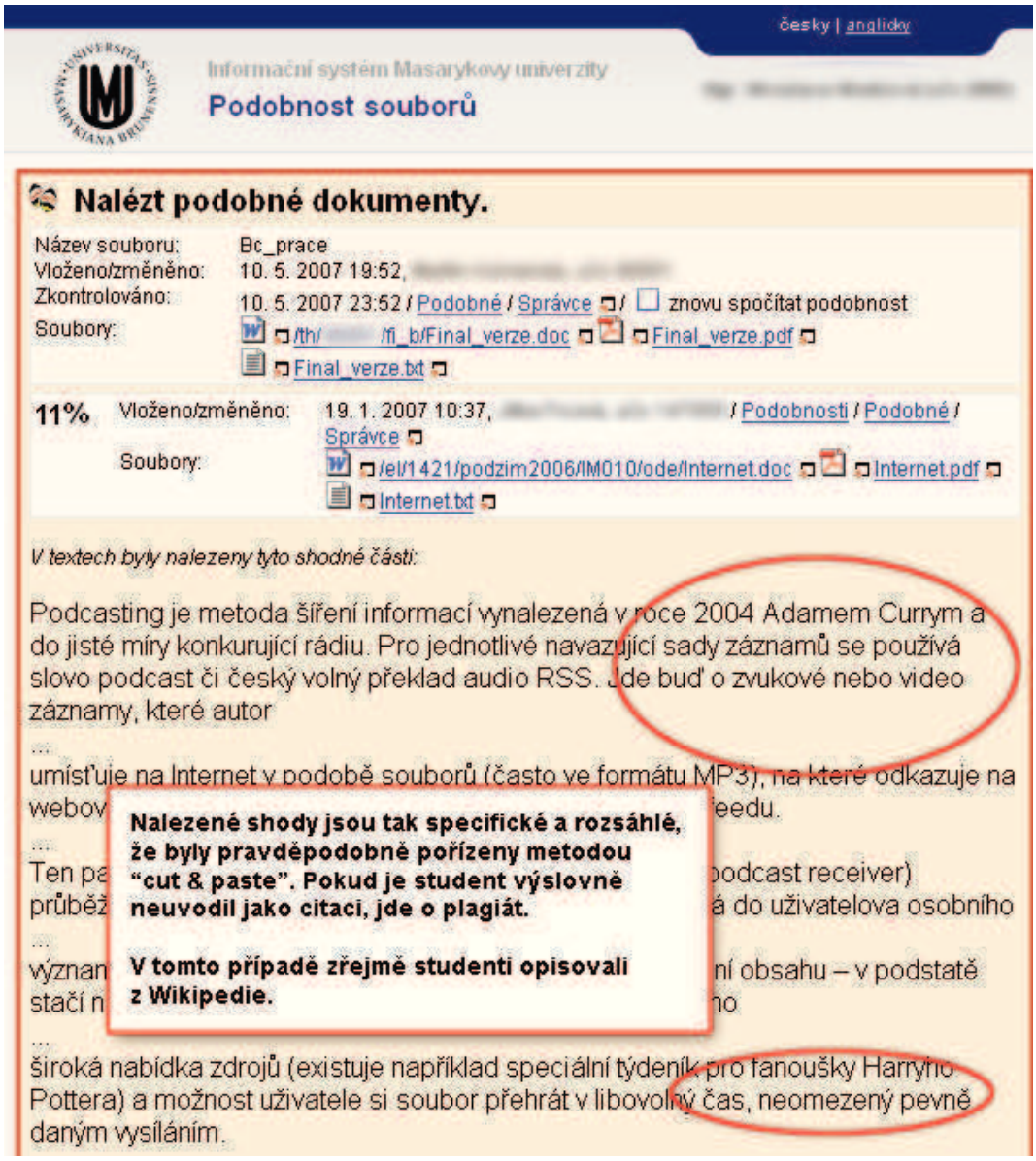


Obrázek 2: U nalezených obdobných dokumentů klikne na Podobnosti

za synonyma apod. Pro tyto účely by systém musel disponovat obrovským výpočetním výkonem, aby mohl porovnávat práce na takovéto úrovni přes celý repozitář. Využívá tedy metody ukládání „metadat“, která jsou ke každé práci vytvořena. Jedná se o sekundární údaje, které vzniknou zpracováním textu obsaženém v práci a ty jsou mnohem více optimalizovány pro vyhledávání a porovnávání. Jako metadata si můžeme představit informace, mezi které patří autor, datum vytvoření, klíčová slova apod. Systém je neustále vyvíjen a nabízí poměrně dost funkcí a rozšíření:

- vyhledávání v již zmíněných metadatech,
- správování uživatelů a dat,
- fulltextové vyhledávání,
- tématické vyhledávání,
- vyhledávání podle kritérií:
  - procentuální podobnost,
  - podle stáří prohledávaných prací,
  - podle repozitáře(v rámci univerzity a jednotlivých fakult).
- hromadný import prací nebo individuální import samotnými studenty,
- evidenci posudků k závěrečným pracem,
- rozšíření o internetové zdroje.

IS dnes obsahuje desítky tisíc závěrečných prací a tento počet není finální, protože neustále přibývají nové a nové. Samy o sobě tvoří zlomek porovnávaných dokumentů, protože systém neporovnává práce pouze mezi sebou, ale porovnává je i s ostatními



Obrázek 3: Vypíší se shodné úseky textu - učitel posoudí, zda jde o opisování

dokumenty, publikacemi, skripty a jinými dokumenty. Jejich počet by se dal odhadnout na řádově na miliony.

Do projektu je aktuálně zapojeno 36 českých veřejných a soukromých vysokých škol a několik zahraničních vysokých škol. Každá z těchto institucí má k dispozici vlastní konfiguraci systému a může si jej nakonfigurovat podle svých požadavků. Nastavit specifická práva pro uživatele, pro jednotlivé práce. Hlavní myšlenkou a cílem je zapojit všechny vysoké školy do boje proti plagiátorství. Ale není to jednoduchý úkol, jelikož s velkým množstvím prací, informací a dokumentů může docházet ke konfliktům mezi univerzitami. Jedním z důvodů by mohlo být porušování autorských práv, protože některá díla není úplně jednoduché sdílet ani v rámci detekce plagiátů. Tím, že systém pracuje s citlivými daty, nesmí sám o sobě porušovat autorský zákon.

Vezmeme-li v úvahu původní myšlenku, tak je jasné, že se postupně naplňuje, ale odhalování plagiátů stále nebude stoprocentní, protože může „jednoduše“ docházet ke kopírování prací z univerzit, které momentálně nejsou součástí centrálního registru. Pokud porovnáme objem dat, které tento registr obsahuje, tak se jedná o relativně nízké procento [21].

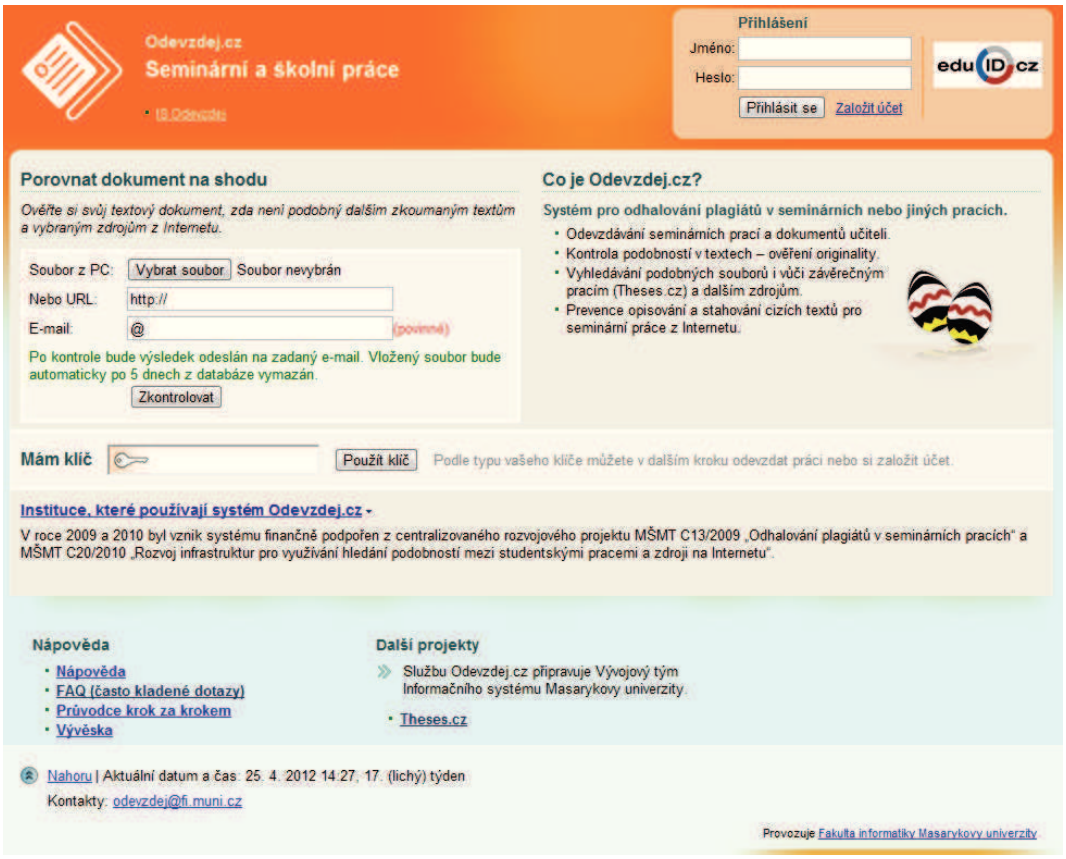
2.4.3 Dceřiný projekt Odevzdej.cz

Projekt Odevzdej.cz zobrazený na obrázku 4 je vyvíjen rovněž Masarykovou univerzitou v Brně, proto se dá považovat za dceřiný. Důvod jeho vzniku je opodstatněn jak ze strany pedagogů tak i ze strany veřejnosti. Pedagogičtí pracovníci upozorňovali na to, že studenti nekopírují pouze při ukončování studia, ale i v jeho průběhu. Veřejnost nejen tedy studenti, chtěli mít k dispozici systém, který by využíval repozitáře systému Theses.cz aby mohli jednorázově otestovat vloženou práci, aby zjistili, zda se třeba nevědomky nedopouštějí plagiátorství.

Vývojové oddělení Masarykovy univerzity se rozhodlo, že systém zpřístupní všem. To bylo totiž i původní myšlenkou, že systém Theses.cz bude kontrolovat nejen závěrečné práce, ale i jednotlivé semináry apod. Ovšem v prvopočátcích se projevily tzv. „porodní bolesti“, protože bylo komplikované vytvořit tak komplexní řešení. Proto pro tyto účely vznikl samostatný projekt do nějž se zapojilo několik univerzit, které se zapojily do centralizovaného rozvojového projektu MŠMT s názvem „Systém na odhalování plagiátů v seminárních pracích“.

Odevzdej.cz umožňuje vyhledávat plagiáty, jak mezi závěrečnými pracemi, tak i mezi semestrálními pracemi, referáty, laborarními protokoly, slohovými cvičeními a dalšími pracemi, všude, kde je potřeba ověřit původ a originalitu. Je zpřístupněn nejen vysokoškolským pedagogům, ale i středoškolským profesorům a ostatním soukromým osobám(není zde nutná autentizace), kteří si chtějí ověřit, zda-li daný dokument není plagiátem. Postačí, když do systému nahrají příslušný soubor a systém jim po jeho převodu do formátu PDF a analýze textu po určité době nutné pro zpracování zašle výsledky do emailové schránky [8] [9].

Nespornou výhodou takového systému je testování podezřelých souborů na více místech. Může se jednat o lokální úložiště škol zapojených do projektu, tak i o efektivnější způsob než při použití vyhledávacích služeb jako Google nebo Bing, protože ne všechny



Obrázek 4: Hlavní strana systému Odevzdej.cz

Zpracovávaný dokument	**pozitivní shoda**
Název dokumentu:	doc_nul000_2011_wiki2.pdf
Vloženo od:	46.47.172.246
Vloženo:	25. 4. 2012 16:45, nepřihlášený uživatel
Podobné dokumenty:	
INFO:	Soubor pochází z jiného systému: is.vstecb.cz
Podobnosti:	https://odevzdej.cz/auth/dok/plag_podobnost_uzlu.pl?u1=527b6e65295ad41e;u2=d65244094a9b01e5;s1=9;s2=10
Míra shody:	68 %
Název dokumentu:	Wikpedia.org
Podobnosti:	https://odevzdej.cz/auth/dok/plag_podobnost_uzlu.pl?u1=527b6e65295ad41e;u2=d78b07a0271c85fc;s1=9;s2=9
Míra shody:	22 %

Tabulka 3: Systém Odevzdej: Informace o podobnostech vašich souborů, pozitivní shoda

Zpracovávaný dokument	**negativní shoda**
Název dokumentu:	val111_2011system_matprac__2_.pdf
Vloženo od:	46.47.172.246
Vloženo:	25. 4. 2012 16:47, nepřihlášený uživatel
Výsledek porovnávání dokumentu	
K vloženému souboru nebyl v databázi nalezen žádný podobný dokument.	

Tabulka 4: Systém Odevzdej: Informace o podobnostech vašich souborů, negativní shoda

školy umožňují těmto nástrojům přístup ke svým databázím. Další výhodou je přístup k rodičovskému projektu Theses.cz a to konkrétně k celému repozitáři závěrečných prací a všem dokumentům, které obsahuje. Také je schopen porovnávat soubor vůči známým a ověřeným informačním systémům, které působí jako zdroj relevantních informací např. k otevřené encyklopedii Wikipedia. Nabízí také službu v podobě úložiště, které v podobě e-learningového prvku mohou studenti odevzdávat své práce, které jsou pak automaticky porovnány a učitel si je zde může také vyzvednout. Jedná se o rozšířenou službu poskytovanou především školám, zapojených do projektu odevzdej.cz [8] [9].

2.5 Systémy vyvinuté na VŠB

2.5.1 Nástroj pro identifikaci plagiátů a podobných dokumentů

V diplomové práci [24] se autor zaměřil na dva aktuální problémy a to na identifikaci plagiátů a nalezení podobných dokumentů. Účelem bylo zaměřit se, jak na výzkumnou část, tak i na aplikační charakter práce. Důležitou součástí bylo nastudovat a realizovat práci s datovými kontejnery, archívy, různými datovými formáty a typy souborů, tzn. že práce měla využívat lokálního úložiště, respektive lokálního repozitáře, kde se zpracovávané dokumenty budou ukládat, indexovat a poslouží k vyhledávání plagiátů a podobností. Seznamuje čtenáře s problematikou plagiátorství, s různými metodami zpracování textu od SVD metody, přes váhové metody až po určení podobnosti textu. Hlavní rozdíl práce je v tom, že nevyužívá externích vyhledávacích služeb pro hledání obdobných dokumentů, ale zabývá se především více metodami analýzy textu s využitím repozitáře. Výstupem je množina dokumentů, kterým se plagiovaný dokument podobá.

2.5.2 Návrh repozitáře studijních projektů se zaměřením na rozpoznávání podobných prací

Předmětem bakalářské práce [25] jak už z názvu vypovídá, bylo navržení repozitáře, který by sloužil k ukládání projektů a zároveň by umožňoval rozpoznávání podobných prací a upozornil by na to, pokud některý student opisoval od jiného studenta. Umí zpracovávat základní údaje o vstupních dokumentech včetně jejich textového obsahu. Využívá při tom algoritmu *String similarity* a indexování, kde indexování probíhá v podobě tvorby slovníku pro každou jednotlivou práci s četností jejich výskytu. Výsledné tabulky se pak

porovnávají mezi sebou. Oproti tomu algoritmus *String similarity*, vyhodnocuje podobnost textů na základě výskytů řetězců a vzdálenosti mezi nimi. Opět se jedná o práci využívající lokální uložiště pro porovnávání prací.

3 ANALÝZA A ZPRACOVÁNÍ TEXTU

Při dnešních možnostech sdílení informací, jejich distribuci, vzdělávání a především díky internetu je poměrné snadné dostat se k čemukoliv. Tohoto faktu jsme si vědomi a také víme, že se díky němu stále častěji potýkáme se vznikem plagiátů. Nesmíme ale zapomínat, že autoři těchto padělků jsou také chytří a mají přinejmenším základní ponětí o tom, jak lze takové kopie detekovat. Dokážou poměrně snadno obejít metody „copy-paste“ ať už se jedná o kopírování vět, celých odstavců, nebo kompletních bloků bez jakýchkoliv úprav. Musíme tedy vzít v úvahu sofistikovanější metody pro zjišťování takto zamaskovaných textů, která práce je plagiát a která není, a nesmíme se spoléhat pouze na to, že detekce „copy-paste“ bude dostatečná.

Nejdůležitějším bodem, na kterém stojí celá metoda odhalování plagiátorství je postup, při kterém se z podezřelého souboru vybere vhodná množina dat, která slouží pro vyhledávání obdobných dat, které mohou potvrdit jestli se jedná o plagiát nebo ne. Jinak řečeno, pouze tato množina vstupů bude sloužit jako potencionální měřítko (záleží tedy na objemu těchto vstupních dat), které se budou srovnávat nejen s testovaným souborem, ale také mezi sebou. Nikdy ne mezi více, pokud danou množinu nezvětšíme.

Před tím, než stanovíme pravidla, která budou určovat, jak s jednotlivými texty pracovat, jak jej korektně porovnávat, musíme se podívat hlouběji na to, z jakých prvků je vstupní množina tvořena. Kapacita a výkony výpočetních strojů jsou více než dostatečné a poměrně snadno dokáží porovnat dva textové soubory, ale v základním principu nám zodpoví pouze otázku úplné shodnosti vstupních souborů. Považujme to tedy za základní princip, který používají všechny nástroje pro detekci podobnosti. Svým způsobem testují shodu dvou řetězců z odlišných vstupů. Záleží na vstupní granularitě, může se jedna o slova, věty, souvětí, odstavce, ale i celé stránky. Velikost této množiny a velikost takového řetězce nám přímo ovlivňuje dva faktory při zpracování textů.

Hlavními kritérii je přesnost a rychlost výpočtu. Použité algoritmy se dají rozdělit do dvou skupin, ta první porovnává řetězce na úrovni jednotlivých znaků, oproti tomu ta druhá provádí porovnávání na úrovni jednotlivých slov [1] [2] [3].

3.1 Přímé porovnávání řetězců

Předpokladem pro to, abychom mohli použít algoritmy z kterékoliv z těchto dvou uvedených skupin, je stejná délka obou testovaných řetězců. Jedním z nich je porovnávaný řetězec neboli podezřelý, který porovnáváme proti potencionálně zdrojovému (původnímu). Vezměme v úvahu modelovou situaci, která může nastat. Máme k dispozici vstupní dokument, u kterého potřebujeme ověřit pravost. Porovnávaný text má celkovou délku 5000 znaků, a délku řetězce pro porovnávání zvolíme na 20 znaků, značit jej budeme písmenem l.

Vstupní množina pro porovnání bude obsahovat 4980 řetězců (5000 - l). Porovnáváme celý text, ale po relativně malých úsecích. Pokud bychom chtěli snížit počet porovnávaných řetězců můžeme například zvětšit délku na dvojnásobek tj. l = 40. Ale tohle nám příliš nepomůže, protože počet porovnávaných řetězců klesne jen na 4960. navíc příliš

velkým zvětšováním nebo zmenšováním délky porovnávaných řetězců zavádíme nepřesnosti do porovnávání.

Takže při porovnávání dvou dokumentů, které musíme vzájemně porovnat bychom museli provést obrovské množství operací. Museli bychom porovnat 4980^2 dvojic (24800400 kombinací). A to se jedná pouze o porovnání dvou souborů, naneštěstí porovnání musíme provést na mnohem větším vzorku, protože jak již bylo zmíněno vstupní vzorek nám určuje „měřítko“, v jakém se porovnání provádí. Tzn. výsledný počet dvojic by závisel na objemu dat souborů určených k porovnávání [1] [2].

Příklad 3.1
Kombinace řetězců které vznikají při porovnávání (l = 20) vypadá následovně:

Vstupní text: „Odhalování plagiátů má zabránit zneužívání duševního vlastnictví jiných osob a zamezit studentům vydávat zdrojové kódy jiného člověka za svou práci, což je v rozporu s řádem vysoké školy.“

„Odhalování plagiátů “ - „má zabránit zneužívá“,
„Odhalování plagiátů “ - „á zabránit zneužívá “,
„Odhalování plagiátů “ - „ zabránit zneužívá d“,
„Odhalování plagiátů “ - „zabránit zneužívá du“,

■

Podle tohoto příkladu už si snadněji představíme, že hlavním problémem je přesnost. Je tedy důležité snížit celkový počet kombinací a tím stanovit délku porovnávaného řetězce l. Pokud určíme malé číslo bude počet kombinací růst exponenciálně a přesnost bude klesat, protože nám bude ve výsledku vracet mnoho nerelevantních shod. Pokud zvolíme velké číslo, počet kombinací se nám sice sníží, zvýší se nám i rychlost, ale zase zde zaneseme velkou nepřesnost. Stačilo by kdyby autor pozměnil slovosled a už nebudeme schopni vstupní řetězce porovnat, a my bychom byli schopni detekovat pouze plagiáty typu „copy-paste“, protože množina bude přesahovat přes tyto změny. Navíc při velikosti množiny délky 500 bychom neodhalili text, který je menší než tato délka množiny [4].

3.2 Metoda fingerprint

Princip fingerprintu neboli „otisku prstů“ umožňuje řešit problém ve stanovení délky porovnávaného řetězce délky l. Jeho technika nespočívá v přímém porovnávání samotných řetězců. Nejdříve z každého z nich vytvoří tzv. fingerprint a ty potom následně porovnává. „Fingerprint“ naznačuje, že by se mělo jednat o jedinečné digitální otisky porovnávaných textů. Takových algoritmů existuje nepřeberné množství a proto si nastíníme pouze základní z nich, protože jenom technika fingerprintu by obsahově vydala na samostatnou práci. Takový jednoduchý algoritmus si můžeme představit tak, že ve zpracovávaném textu vynechává samohlásky, mezery a diakritiku [4] [14].

Příklad 3.2
Jednoduchý příklad algoritmu využívajícího ”fingerprint”metody:

Vstupní text: „Odhalování plagiátů má zabránit zneužívání duševního vlastnictví jiných osob a zamezit studentům vydávat zdrojové kódy jiného člověka za svou práci, což je v rozporu s řádem vysoké školy.“

„Odhalování plagiátů má zabránit zneužívání “ - „dhlvnplgtmzbrntznzvn“

„odhalovani plagiatu ma zabranit zneuzivani “ - „dhlvnplgtmzbrntznzvn“

■

Místo přímého porovnávání textů, se tedy porovnávají digitální otisky textů, což nám přináší jisté výhody:

- vynechání nežádoucích znaků při porovnávání jako je diakritika, překlepy, značky případně správná gramatika,
- v nejlepších případech se objem porovnávaných dat zmenší cca o 30%,
- v případě, že použijeme metodu převodu textu na čísel, získáme další datovou úsporu (např. Hoffmanovo kódování jednotlivých znaků nebo LZ kódování skupin znaků).

3.3 Využití n-gramů

Další z metod, které určitě stojí za zmínku, je využití n-gramů, které představují spojitou posloupnost znaků, ať už se jedná o text nebo řeč. Mohou být složeny z jednotlivých fonémů, slabik, písmen, znaků nebo slov v souvislosti s použitím. Základním n-gramem je „unigram“, který má velikost délky jedna, dalším je „digram“ délky dva a další „trigram“ o délce tři. Větší n-gramy jsou pojmenovávány jako „four-gram“, „five-gram“ atd. Pro použití je třeba n-gramový model, který představuje typ pravděpodobnostního jazykového modelu určeného k předpovídání dalších položek v pořadí n - 1. N-gramy se používají při výpočtech pravděpodobnosti, při zpracování přirozeného jazyka a při datové kompresi [6] [7] [13]. V tabulce 5 je zobrazeno, jak n-gram vypadá.

N-gramy se tedy dají použít jako efektivní nástroj pro detekci přibližné shody. Tím, že převedeme pořadí položek v našem případě textový vzorek na soubor n-gramů, tak získáme vektorový prostor, se kterým můžeme dále pracovat. Jeho velkou výhodou je velmi rychlé a efektivní porovnávání s jinými sekvencemi tedy s jinými sobory také převedenými do n-gramů. Můžeme si demonstrovat příklad, ve kterém budeme převádět řetězce do trigramu. Dostaneme trojrozměrný prostor, kde první prostor určuje počet výskytů řetězce „AAA“, druhý „Aab“, a tak dál pro všechny možné kombinace tří znaků. Ztratíme informace o některých řetězcích, protože po převodu nebudeme vědět původní pořadí. Např. řetězec „FGE“ se nám ve výsledku promítne do digramu „fg“ a úplně stejně „EFG“ se nám promítne také do digramu „fg“, což není úplně to samé. Ale z jazykového pohledu je tento problém zanedbatelný, protože zpracováváme reálný jazyk a pak je velmi pravděpodobné, že vektorová prezentace těchto řetězců bude podobná [6] [7].

Obor	Jednotka	Vzorek	1-gram	2-gram	3-gram
Název			unigram	bigram	trigram
Pořadí Markov modelu			0	1	2
Pořadí protejnů	amino, kyselina	...Cys-Gly-Leu-Ser-Trp...	..., Cys, Gly, Leu, Ser, Trp, ...	...,Cys-Gly, Gly-Leu, Leu-Ser, Ser-Trp,...	...,Cys-Gly-Leu, Gly-Leu-Ser, Leu-Ser-Trp,...
Pořadí DNA	základní dvojice	...AGCTTCGA..	..., A, G, C, T, T, C, G, A,...	..., AG, GC, CT, TT, TC, CG, GA,...	..., AGC, GCT, CTT, TTC, TCG, CGA,...
Počítačová lingvistika	znak	...být_či_nebýt	...,b,ý,t,_,č,i,_,n,e,b,ý,t,...	...,bý,ýt,t_,_č,či,in,ne,...	...,být,ýt_,t_č,_či,či_,i_n,..
Počítačová lingvistika	slovo	...být či nebýt to je oč...	...,být,či,nebýt,to,je,oč,...	....,být či,či nebýt,nebýt to,to je,je oč,...	...být či nebýt, či nebýt to,nebýt to je,to je oč,...

Tabulka 5: Příklady n-gramů z různých odvětví

3.4 Redukce kombinací porovnávaných řetězců s využitím n-gramů

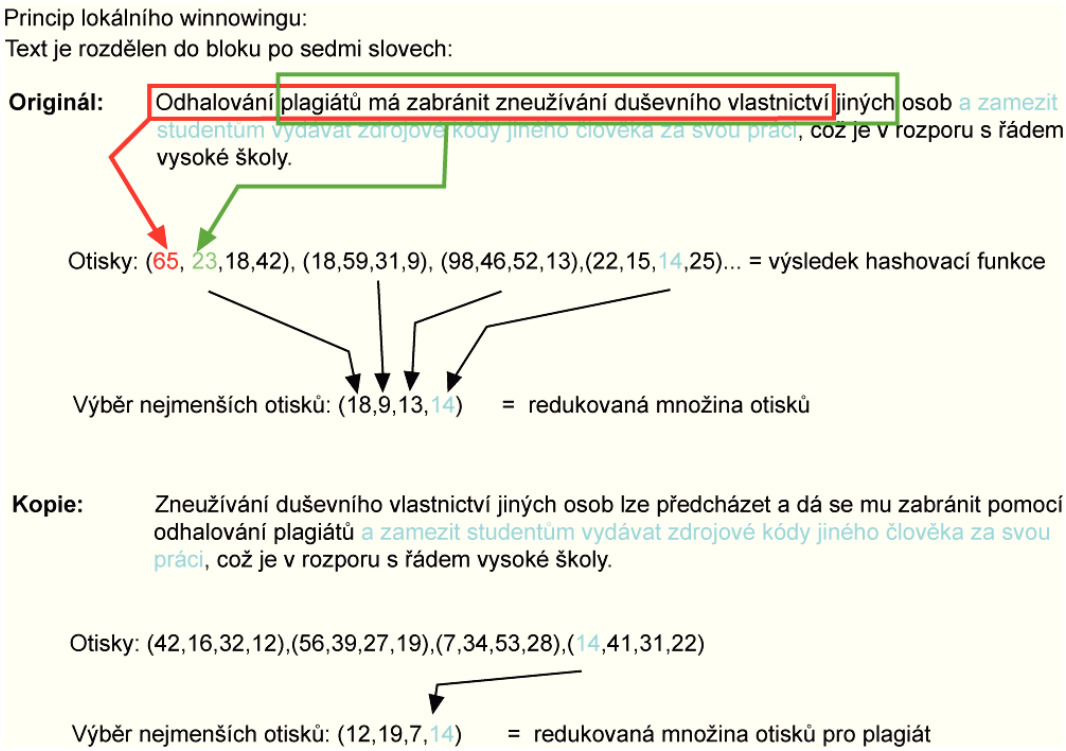
Metoda fingerprintu přináší sice několik vylepšení oproti přímému porovnávání řetězců, ale stále nám neumožňuje vyřešit problém obrovského množství porovnávaných řetězců, které nám vznikají při porovnávání dvou a více dokumentů. Dalo by se jednoduchou metodou vymyslet redukční pravidlo, které by pomohlo snížit počet porovnávaných řetězců. Např. bude se vybírat pouze každý desátý řetězec, ze kterého se vytvoří otisk, ale tento způsob má i své nevýhody, protože tímto náhodným výběrem můžeme pravděpodobně minout část, kterou plagiátoři kopírují.

Na Cornellově univerzitě, přišli s podstatným řešením, které je založeno na porovnávání skupin celých slov, místo porovnávání řetězců. z pohledu přirozeného jazyka má tento způsob jistou logiku, protože člověk nemyslí v řetězcích, ale ve slovech a větách. To nám umožňuje minimalizovat počet kombinací řetězců, protože se zaměřujeme na celé věty a tím snižujeme čas potřebný pro zpracování a zvyšujeme přesnost. Pokud bychom testovaný text rozdělili dále na věty, ze kterých by se vybíraly a porovnávaly některá slova opět by to přispělo ke zrychlení. Po provedení několika testů a měření stanovili optimální počet slov v porovnávaném řetězci na l = 7, s tím, že se porovnává okolí slov obsažených v tomto řetězci o délce w = 4. Laicky řečeno vždy se vybere text, který obsahuje 11 slov a pokaždé je z tohoto textu vybrán text o sedmi slovech, ze kterých je pak vytvořen digitální otisk. Princip porovnávání je tedy poté shodný se základním principem, kdy se posouváme v řetězci po slovech a tím nám vznikají další otisky. A umožňuje nám to přesnější a rychlejší hledání shod [17].

Algoritmus by mohl vypada následovně:

1. Rozdělení textu na věty.
2. Zpracování vět bude probíhat postupně od začátku.
3. Vytvoření vzorků složených skupinou sedmi slov, následující vzorek se posune o slovo dál ze kterých se poté vytvoří otisky. z věty o 30ti slovech by se tak vytvořilo 30 - l + 1 otisku tj. 22 vzorků.
4. Využití ”winnowing”algoritmu, který ze těchto skupin otisků, vybere úseky, které se nepřekrývají a u každého úseku vybere nejmenší otisk.
5. Vybrané úseky se poté uloží do systému s odkazem na původní dokument a s jejich pomocí se provede porovnání dokumentu.

Z obrázku 5 je možné vidět, jak algoritmus postupuje při vytváření digitálních otisků. Využívá sedmi za sebou následujících slov, z kterých vytvoří otisky a poté je rozděluje do skupin po čtyřech slovech, přičemž v každé skupině se nachází čtyři po sobě jdoucí otisky. v každé skupině je vybrán otisk, který má nejnižší hodnotu a ta se poté uloží do množiny otisků, které se ve finále porovnávají. Až jsou vytvořeny všechny potřebné množiny pro testovaný dokument a samozřejmě pro dokument, s kterým se porovnávání provádí, tak se zkontrolují výsledné množiny otisků. Pokud dojde k nalezení otisku, který má stejnou hodnotu v obou množinách, pravděpodobně se nám podařilo nalézt



Obrázek 5: Princip lokálního winnowingu

důkaz o plagiátorství. U výše uvedeného příkladu by se jednalo u část textu s hodnotou otisku 14, což znázorňuje text: „zamezit studentům vydávat zdrojové kódy jiného člověka za svou práci, což je v rozporu s řádem vysoké školy“. Dá se s jistotou říci, že plagiáty, které budou složeny z jedenácti slov budou bez problému odhaleny (tj. sedm + čtyři slova). Věty délky sedm až jedenáct slov ovšem nemusí být nalezeny.

Metoda lokálního winnowingu ve své podstatě definuje jak postupovat při redukci kombinací s využitím n-gramů. v ukázkovém příkladě se jedná o four-gramy, které představují vektorový prostor délky 4. Hlavní výhodou této metody je výrazná redukce kombinací, které jsou potřeba k porovnávání a také determinismus, kterým je zajištěno, že zkopírovaný text o x slovech nacházející se kdekoliv v textu bude nalezen. Za nevýhodu můžeme považovat složitější zpracování, kdy si musíme ke všem otiskům ukládat informace o tom, k jaké části věty patří, ke kterému sousloví.

Lokální winnowing algoritmus je pouze jedna z používaných metod. Existují i další neméně sofistikované metody využívající například sémantické analýzy. Mezi nejznámější patří metoda „Séman“, jejíž princip je založen na překladu slov do univerzálního kódu. Představme si hashovací funkci, která vždy pro stejné slovo vrací právě jeden a pokaždé stejný výsledek. Výhoda ovšem spočívá v tom, že podobným slovům stejného významu je přiřazen stejný kód, tzn. že například slovům podobným, nebo synonymům bude přiřazen stejný kód, což umožňuje detekci i mezi mírně pozměněnými texty. S pomocí sémantické analýzy se dá vytvořit silný a robustní nástroj pro odhalování a detekci plagiátorství [5] [2] [1].

4 NÁVRH SYSTÉMU

Vývoj každé aplikace ať už nové, nebo dodatečné rozšiřování starší aplikace, přidávání funkcionality, změny funkcionality apod. nezávisle na samotné náročnosti a komplexnosti se v dnešní moderní době řídí vhodnými pravidly a postupy. Jedná se o dodržování metodik, které samotnému vývojáři, týmu, oddělení ujasní a zpřehlední kompletní proces vývoje a tím zjednoduší možná budoucí rozšíření. Bavíme se zde o metodikách spadajících do kategorie „Unified Process".

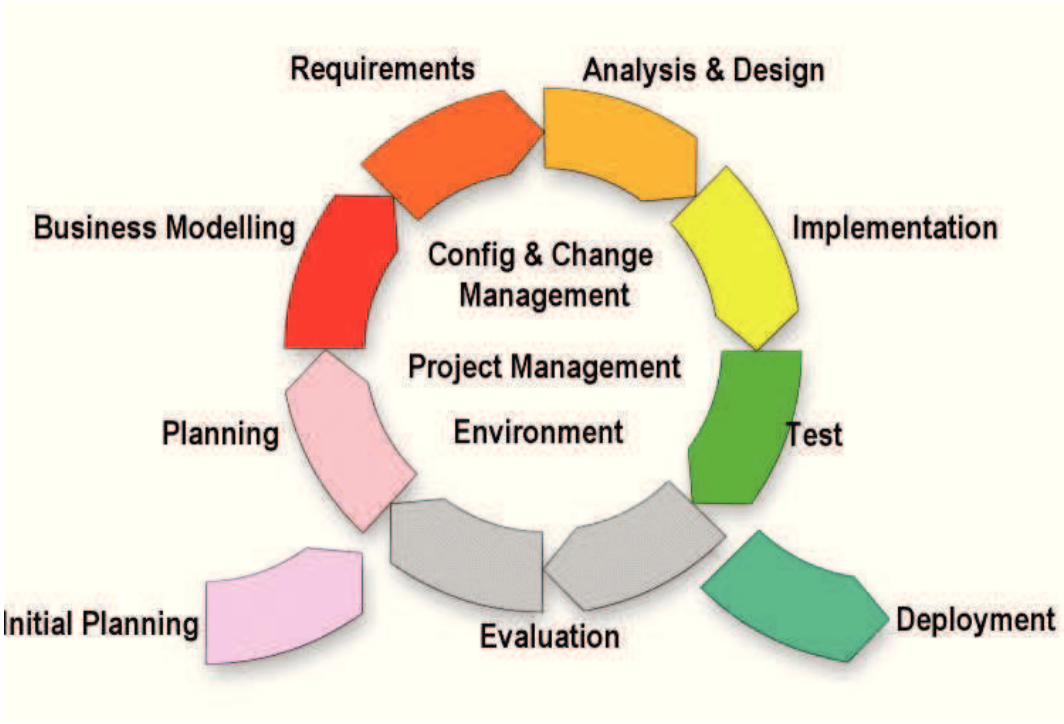
Účelem této metody je rozdělení vývoje aplikací do několika hlavních fází: plánování, sběr požadavků, analýza a design, návrh implementace, testování, revize. Těchto šest hlavních aktivit tvoří seskupení, které můžeme v průběhu vývoje opakovat, je tedy označováno jako iterace. Každá aktivita samostatně je více, či méně aplikována do jednotlivých fází vývoje: zahájení, rozpracování, konstrukce a nasazení do provozu, kde každá z těchto fází může obsahovat libovolné množství těchto iterací v závislosti na komplexitě projektu [12].

V následujících subkapitolách se budeme zabývat aktivitami spadajícími do fáze zahájení a rozpracování tj. především sběrem požadavků, analýzou a návrhem.

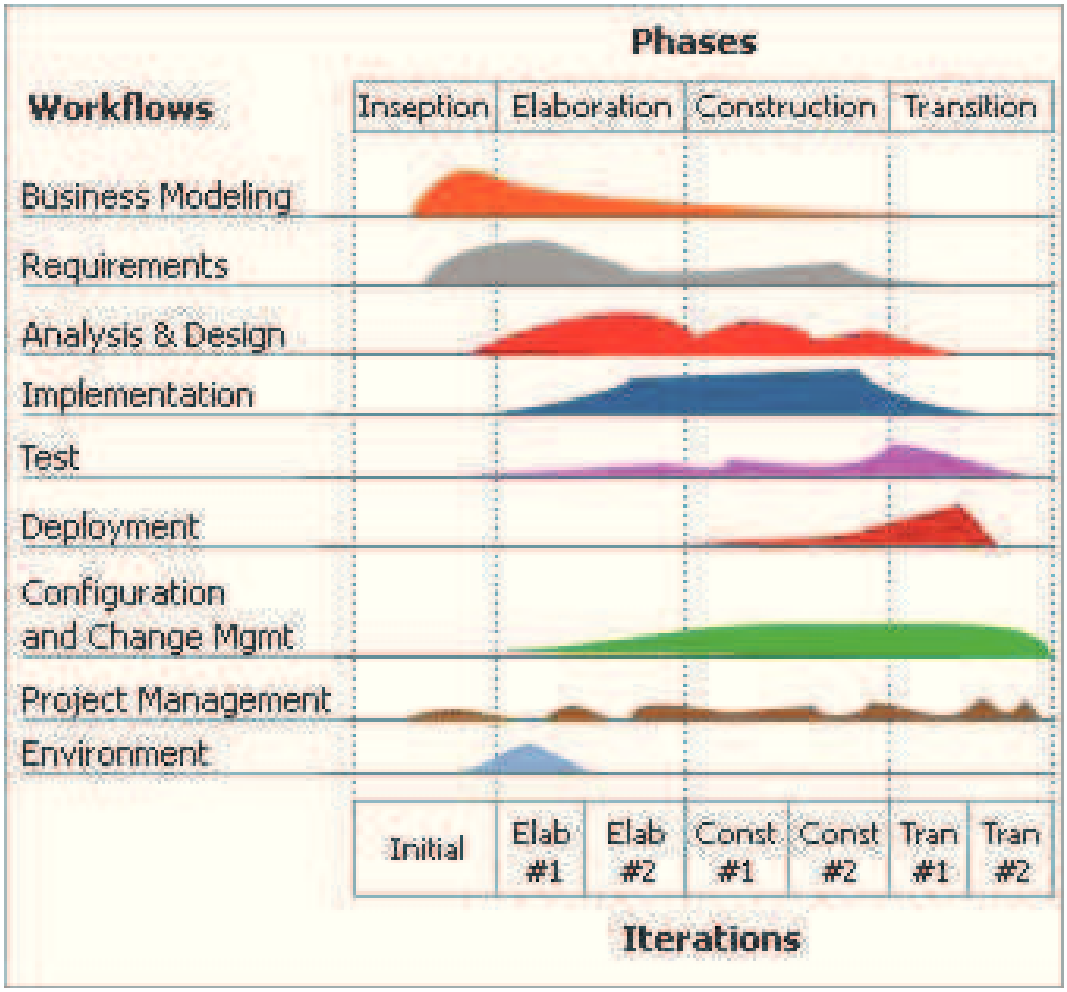
4.1 Účel systému

Důležitým prvkem vývoje jakéhokoliv nového produktu, nejen tedy softwarového, ale jakéhokoliv z kterékoliv jiné oblasti, je podmíněn základním faktorem. Tím je motivace. Před samotným začátkem si musíme položit plno otázek, které mají motivační charakter: „Co nám produkt přinese? Jaký bude mít smysl? Přinese nám nějaký zisk? Kdo je cílová skupina a pro koho je produkt určen?", přičemž ne vždy se nám podaří najít úplně reálné odpovědi, protože to není jednoduchý úkol. Může se jednat o malé přínosné věci jako zautomatizování ručního procesu, který nám ušetří čas, zefektivnění sběru dat, optimalizace výpočtů apod. Položme si tedy několik otázek a důvodů k systému, který má vzniknout: „Proč potřebujeme disponovat aplikací pro odhalování plagiátů? Co nám tedy přinese vytvoření takové aplikace? Proč nevyužijeme již existujících služeb např. Theses.cz , které detekci plagiátů nabízejí?".

Hlavním důvodem je, jak již bylo zmíněno výše, vytvořit aplikaci, která nám umožní získat zpětnou vazbu nad zpracovanými pracemi středoškolských studentů a tím, jestli se dopouštějí plagiátorství nebo ne. Vytvoření aplikace nám přinese nejen přehled o jednotlivých závěrečných pracích, které jsou součástí prvotního konceptu, ale také v pozdějších fázích vývoje nabídne doplňující služby, které postupem času vyplynou z užívání této aplikace. Důvodem, proč nevyužijeme existujících služeb je, že i závěrečné práce středoškolských studentů musí být chráněny autorským zákonem a nebylo by tedy možné jednoduše upravit legislativu tak, abychom mohli nahrát dané soubory do systému třetích stran a tam spustit porovnávání. Další důvodem, proč vzniká vlastní řešení je, že si budeme sami určovat, kam se bude systém vyvíjet. Stručně řečeno výsledkem bude aplikace, která bude mít preventivní charakter, bude poskytovat informace o potencionálním plagiátorství a bude poskytovat informace pro zlepšení kvality středoškolské výuky. Samozřejmě vždy bude na zodpovědném pedagogovi, jak se k problému plagiátorství



Obrázek 6: Vztah jednotlivých fází spadajících do jedné iterace



Obrázek 7: Vztah hlavních aktivit tvořících iterace a fáze v Unifikovaném Procesu

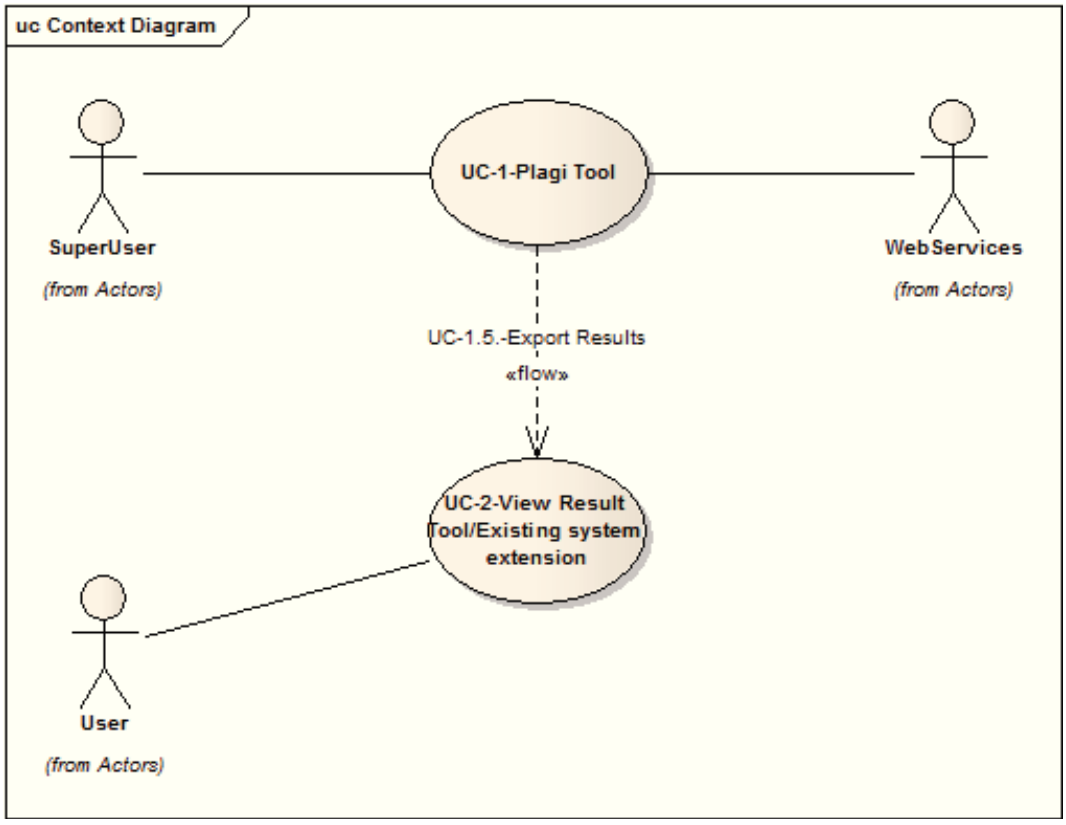
postaví, ale určitě se nebude jednat o systém, podle kterého budou studenti automaticky vylučováni.

4.2 Požadavky

Jak už je uvedeno v samotném zadání této práce, cílem je navrhnout a implementovat systém pro odhalování plagiátů s vyhledáváním shod v internetových zdrojích, tedy vytvořit softwarový produkt, který bude mít potřebnou funkcionalitu. Podstatným faktem je, že systém má využívat již existujícího systému na evidenci maturitních prací a rozšířit jej o zpracované výsledky, tedy nebude přímým rozšířením funkcionality systému, ale pouze mu poskytne zpracované informace. Důvodem je náročnost samotné aplikace především co se dotazů na vyhledávací služby týče, ale také i to, že již existující systém je napsán v odlišném jazyce a běží přímo na webovém serveru školy, na kterém si nemůžeme dovolit spouštět takto výpočetně náročnou aplikaci. Proto systém vznikne jako samostatné nejlépe konzolová aplikace, která umožní snadné rozšíření o další funkce a zpracované výsledky exportuje do databáze již existujícího systému, který se rozšíří o potřebnou databázovou strukturu a několik webových stránek, které umožní zobrazení výsledků.

Základní funkce se dají shrnout do několika bodů:

- systém umožní zpracování soborů PDF s předem definovanou šablonou s strukturou názvů v podobě doc_loginstudenda_rokabsolvovani_casoverazitko.pdf v textové podobě,
- z názvu souboru použije login studenta, který poté použije jako vazbu pro výstupní data,
- zpracování a detekce klíčových slov,
- vyhledávání klíčových slov prostřednictvím jedné z internetových služeb,
- použití relativně jednoduché statistické metody pro porovnávání plagiátů, která nebude vyžadovat dodatečné ukládání dočasných informací,
- uživatelem bude pouze jedna role,
- systém informuje uživatele po ukončení zpracování,
- umožní jednoduchou rozšiřitelnost, podle dalších požadavků, které se objeví v průběhu používání,
- spustitelná odkudkoliv,
- jednoduše konfigurovatelná(vstupní data a výstupní databáze),
- výstupní databáze bude realizována v podobě tabulek MySQL.



Obrázek 8: Diagram zobrazující interakci mezi aktéry a systémem

Na obrázku 8 je jasně znázorněna vazba mezi uživatelem systému pro detekci plagiátorství, webovou službou a vazbou na již existující systému pro správu maturitních prací a jeho uživatelem.

Diagram na obrázku 9 zobrazuje vazbu mezi uživatelem systému (jeho hranici tvoří obdélník) a také vazbu mezi systémem a vyhledávací službou, která přijímá požadavky na vyhledávání a zároveň vrací nalezené výsledky do systému.

4.3 Fáze vývoje

Tento nově vznikající systém můžeme rozdělit do několika vývojových stupňů nebo-li fází, které lze chápat, jako cestu vzniku od nejzákladnějších funkcí, které si uživatel může samostatně využít až po plně automatizování procesu, kdy je vše řízeno bez potřeby zásahu uživatele. Seznam těchto fází má umožnit základní pohled na jednotlivé stupně a potřeby i z hlediska pozdějších inovací a rozšíření.

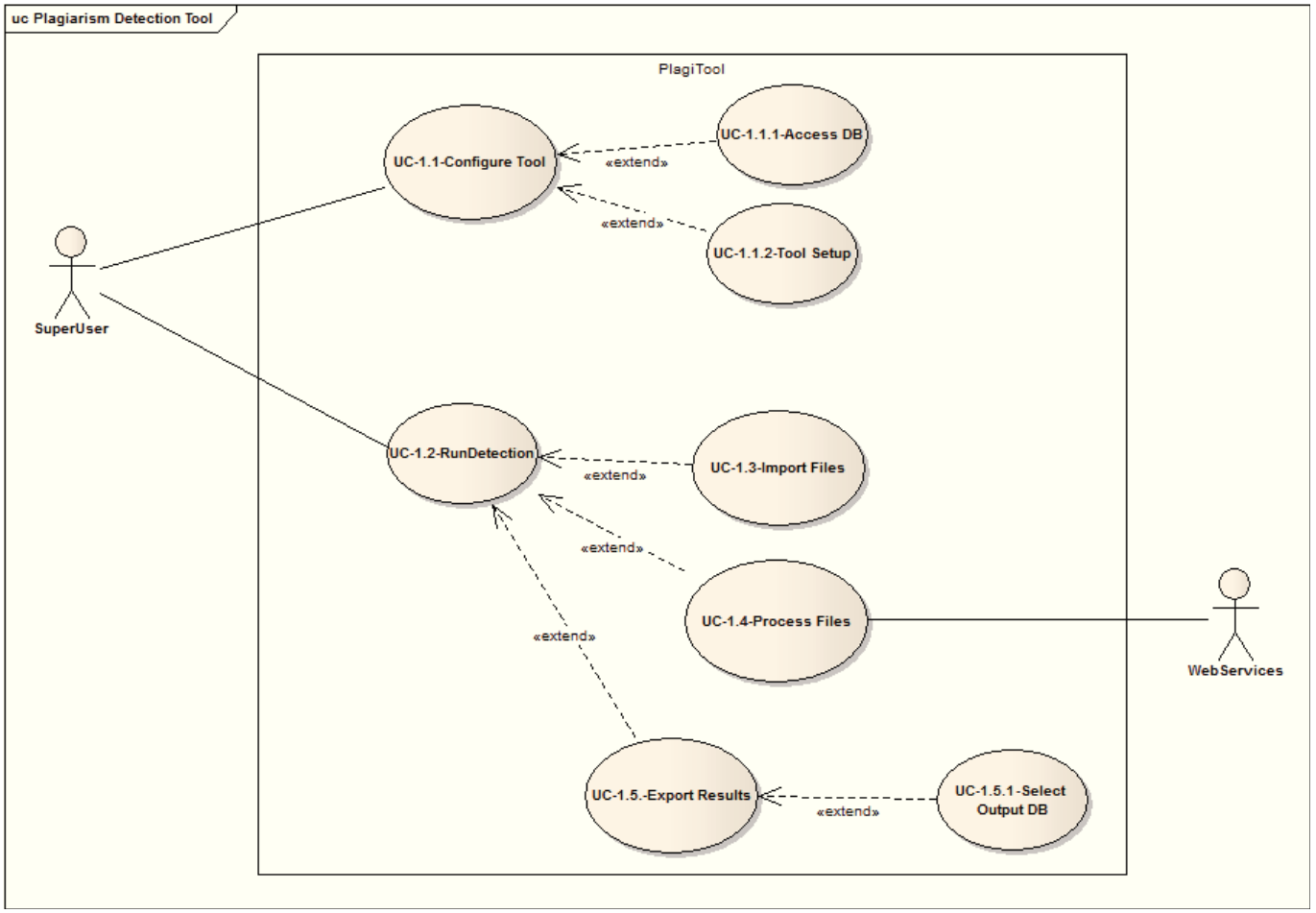
1. **Vyhledávací služba** - můžeme si zde představit využití vyhledávacích služeb jako Google, Bing Yahoo apod., s trochou nadsázky, lze tvrdit, že se jedná o nejprimitivnější nástroj pro vyhledávání plagiátorství na Internetu. Nejprimitivnější proto, že neposkytuje potřebný komfort pro naše použití, tedy nedisponuje zpracováním vstupních textů do podoby klíčových slov, nedisponuje žádným nástrojem pro porovnávání z pohledu uživatele, to ovšem není jeho primární účel. Nicméně se dá využít jako vhodný základ pro náš nástroj.

2. **On-line detekční systémy** - již existující nástroje, určené pro odhalování plagiátorství využívající internetu. Obsahují jednoduché uživatelské rozhraní s dvěmi vstupními poli (jedno pro zdání cesty k porovnávanému souboru, druhé pro přímé vložení testovaného textu). Navíc oproti běžným vyhledávacím službám využívají jistých algoritmů a metodik k sestavování dotazů pro vyhledávání. Po spuštění porovnávání obvykle zobrazí výsledky vrácené přímo z vyhledávače. Takže disponují metodikou detekce klíčových slov.

3. **Pokročilé nástroje** - nástroje přizpůsobené pro větší komfort uživatele. Podporují mnoho typů vstupních formátů, nabízejí několik typů analýz pro zpracování text tj. více metod pro vytvoření vyhledávaného dotazu a také využívají více než jednu vyhledávací službu. Navíc často umožňují porovnávání obsahu jednotlivých dokumentů s nalezenými výsledky.

4. **Plně automatizované nástroje** - z pohledu odhalování plagiátorství se jedná o nejvyšší úroveň aplikace. Celý proces je zautomatizován a nevyžaduje žádný zásah uživatele, pouze minimální v případě, kdy podle zpracovaných výsledků uživatel označí dokument jako plagiát nebo ne. Takto automatizovaný proces si můžete představit jako seznam následujících úkonů:

   (a) Studentům je zpřístupněno úložiště pro odevzdávání prací.



Obrázek 9: Diagram případů užití (use-case)

(b) Systém sám spustí zpracování souborů po té co zjistí přítomnost nových, případně v daných časových intervalech.

(c) Systém načtené soubory zpracuje a zkontroluje jejich obsah.

(d) Systém nalezené výsledky o podobnosti uloží a zobrazí je vyučujícímu.

(e) Vyučující pak tento výsledek ověří z dostupných informací a může označit práci jako plagiát.

4.4 Použité technologie

V následující části se budeme věnovat funkčnosti nového systém a také použitým technologiím pro jeho vývoj. Je potřeba navrhnout vnitřní a vnější strukturu aplikace, nastínit spojení s vyhledávací službou, popsat princip jak aplikace bude fungovat a v neposlední řadě definovat datový model.

4.4.1 Koncept

4.4.1.1 Multiplatformní aplikace Mezi základními požadavky na vznik aplikace je, aby bylo možné ji spustit odkudkoliv. Proto se programovací jazyk Java dá považovat za nejvhodnější volbu, protože k jeho spuštění na jakékoliv platformě postačí nainstalovat vhodný balík s podporou Java Virtual Machine, která nám toto umožní. z hlediska rozšíření již existujícího systému, který je dostupný v podobě webových stránek, by bylo správné tuto funkcionalitu do něj zakomponovat. Ovšem původní systém je napsán v jazyce PHP a běží na webovém serveru školy, který zajišťuje i jiné služby a z tohoto důvodu, nám vzniká omezení, že nemůžeme aplikaci spouštět přímo na tomto serveru. Vznikne nám tedy samostatně běžící aplikace, která bude zpracované výsledky nahrávat do předem definované databázové struktury, o kterou rozšíříme již existující systém. Pokud se systém osvědčí v praxi, tak je vysoce pravděpodobné, že se bude rozšiřovat jeho funkcionalita a budou vznikat časté změny a aktualizace, pak se pro jeho potřeby vyhradí samostatný stroj, na kterém bude aplikace běžet, protože bude vhodné, aby zpracování probíhalo centrálně a umožnilo přístup většímu počtu uživatelů. Systém tedy v počátcích bude existovat jako samostatně spustitelná aplikace a později se zaintegruje do infrastruktury školy a bude zpřístupněn v podobě webové aplikace. Tj. že uživatel zadá pouze potřebnou webovou stránku do prohlížeče, přihlásí se do systému a bude s ním moci pracovat. Nebude zatížen žádnou instalací, aktualizacemi apod. Tzn. že aplikace bude vždy multiplatformní.

4.4.1.2 Použité databáze a slovníky Návrh systému počítá s využitím externích knihoven a slovníku, pro základní zpracování a normalizaci textu. Za tímto účelem využije Postgre SQL databáze a její možnosti použít fulltextové slovníky, které mají schopnost, vracet informace o jednotlivých zpracovávaných slovech. Jaké slovníky jsou použity, jestli se jedná o stop znak (především interpunkční znaménka, spojky, předložky apod.) a také vrací slova v jejich kořenovém tvaru nebo-li „lexému“. MySQL databáze bude určena pouze pro uložení výsledku, které rozšíří právě používaný systém správy maturitních

prací. POStgre použité při zpracování, včetně slovníků pro lexemizaci, MySQL databáze pro export spočítaných výsledků.

4.4.1.3 Extrakce textů z PDF

Pro zpracování PDF souborů bude aplikace využívat knihovny Apache PDFBox v podobě jar balíčků, který po importu do zdrojových kódů aplikace umožní využívat její rozhraní pro načtení a základní zpracování textů. Tzn. že načte a zpracuje pouze textovou část vstupního souboru, všechny ostatní informace jako obrázky, seznamy a metada uložená v PDF souboru bude ignorovat.

4.4.1.4 Detekce klíčových slov

Metoda RAKE - Rapid automatic keyword extraction byla vyvinuta za účelem extrémně efektivní extrakci klíčových slov tak, aby byla použitelná na jakoukoliv množinu dokumentů, i na ty, které ne vždy dodržují gramatiku a navíc jej lze použít pro jakýkoli jazyk. RAKE je založen na pozorování, že klíčová slova neboli vzory obsahují několi slov, ale zřídka obsahují interpunkční znaménka a „stop slova“ jako spojky, zájmena apod. Taková slova jsou z analyzovaného textu vypuštěna, protože nejsou považována za důležité. Je to oddůvodněno tím, že slova, která slouží pro vyhledávání v textu se vyskytují ve větším měřítku, navíc ve spojení s jinými slovy a stop slova neobsahují. Zjednodušeně řečeno jedná se klíčové vzory, které by si vybral běžný uživatel pro hledání obdobných textů.

Vstupem do RAKE metody je seznam stop slov, seznam oddělovačů a samotný text, který bude podroben analýzou. Stop slova a oddělovače se použijí jako oddělovače dokumentu a vytvoří se tak potencionální kandidáti na klíčová slova. Princip spočívá na tom, s jakou frekvencí se vyskytují slova samotná a jak často se vyskytují ve spojení jiných slov. To každému klíčovému slovu zvyšuje hodnotu významnosti, tím větší hodnocení, tím větší význam klíčového výrazu. Nejdříve se tedy algoritmem ohodnotí samotný výskyt jednotlivých slov a poté se ohodnotí výskyt ve spojení s ostatními slovy. z těctho hodnot se poté spočítá váha jak samotných slov tak i klíčových slovních spojení. Slova s nejmenším počtem výskytů a slova samostatná mají nejnižší hodnocení. Slovní spojení si vzájemně zvyšují váhu, a tím jsou pro nás významnější pro hledání obdobných informací než pouhé vyhledávání po jednotlivých slovech.

Samotná přesnost algoritmu je přibližně 70% a autoři uvádějí, že je vhodné a zároveň dostatečné použít pro vyhledávání obdobných textů třetinu zjištěných slov. Tj. pokud nalezneme v textu 30 klíčových slov, na zpracování postačí, když použijeme 10 s nejvyšší váhou. Tato metoda se dá navíc rozšířit o slovník podle zpracovávané kategorie. Tzn. když budeme vědět, že zpracovaný text spadá do kategorie programovacích jazyků, můžeme využít slovníku, který bude obsahovat slova oborného rázu a bude tak uměle zvyšovat potencionální váhu jednotlivých klíčových slov [1].

Příklad 4.1
Příklad textu zpracovaného metodou RAKE:

Vstupní text:

Informatika (počítačová věda) studuje výpočetní a informační procesy z hlediska hardware i software. v praxi se vztahuje k počítačům a od abstraktní analýzy algoritmů, formálních jazyků atd. pokračuje ke konkrétnějším tématům, jakými jsou programovací jazyky, software a hardware. Informační technologie studují vše, co se týká fungování počítačů po technické stránce. Název je odvozen od slova informace, jelikož počítače nepracují s ničím jiným, než s daty (informacemi).

Potencionální kandidáti na klíčová slova (odstraněna interpunkce a stop slova):

Informatika - počítačová věda - studuje výpočetní - informační procesy - hlediska hardware - software - praxi - vztahuje - počítačům - abstraktní analýzy algoritmů - formálních jazyků - pokračuje - konkrétnějším tématům - programovací jazyky - software - hardware - Informační technologie studují - fungování počítačů - technické stránce - Název - odvozen - slova informace - jelikož počítače nepracují - ničím jiným - daty - informacemi

Ohodnocené výrazy klíčových slov:

abstraktní analýzy algoritmů(9) - Informační technologie studují(9) - jelikož počítače nepracují(7) - studuje výpočetní(6) - počítačová věda(5) - informační procesy(5) - formálních jazyků(5) - hlediska hardware(4) - konkrétnějším tématům(4) - programovací jazyky(4) - fungování počítačů(4) - technické stránce(4) - ničím jiným(4) - počítačům(3) - slova informace(3) - hardware(2) - Informatika(1) - software(1) - praxi(1) - vztahuje(1) - pokračuje(1) - Název(1) - odvozen(1) - daty(1) - informacemi(1)

Vyšší číslo znamená vyšší váhu klíčového slova vhodného pro další vyhledávání. Princip výpočtu ohodnocení je přiblížen na obrázku 10 a obrázku 11. z výsledků je vidět, že metoda RAKE je velmi efektivní i přesto, že někdy rozeznává ne zrovna nejvhodnější klíčová spojení. Jedná se ovšem o malé procento, které lze zanedbat. ■

4.4.1.5 Proces zpracování dat a jejich vnitřní formát

Samotné zpracování dat představuje nejen extrahování textu z předložených PDF souborů, které budou rozděleny na jednotlivé sekce podle přiložené šablony, ale také samotné zpracování a analýzu textu. Pro zpracování textu je využita výše zmíněná java knihovna Apache PDFBox, která usnadňuje jejich načítání, ale také i vytváření a spojování s jinými dokumenty. Spadá do open-source systémů tzn. že ji můžeme volně využít, pokud neporušíme licenční podmínky.

Následně jsou v textu vyhledána klíčová slova, která budou použita pro vyhledání podobného obsahu pomocí webové služby Bing. Hlavním důvodem, proč se budou vyhledávat pouze klíčová slova, je snížení datové náročnosti a nepřímého omezení potřebného času pro porovnávání, než kdybychom vyhledávali všechna slova, která text obsahuje, případně jejich kombinace.

Extrahovaný text je následně převeden do lexémů (je základní jednotka slovní zásoby, která obsahuje množinu všech slov určitého slova nebo slovního spojení). Pro převod je použita Postgre databáze s integrovaným českým slovníkem. v zápětí je převedený text zpracován pomocí metody lokálního winnowingu. Tato metoda má výhodu v tom,

	abstraktní	algoritmů	analýzy	daty	formálních	fungování	hardware	hlediska	informace	informační	informatika	jazyky	jelikož	jiným	konkrétnějším	název	nepracují	ničím	odvozen	počítače	pokračuje	praxi	procesy	programovací	slova	software	stránce	studuje	technické	technologie	tématům	věda	výpočetní	vztahuje
abstraktní	1	1	1																															
algoritmů	1	1	1																															
analýzy	1	1	1																															
daty				1																														
formálních					1							1																						
fungování						2		1												1														
hardware							1																											
hlediska							1	1																										
informace									2																1									
informační										2													1					1		1				
informatika											1																							
jazyky					1							1																						
jelikož													1				1			1														
jiným														1				1																
konkrétnějším															1																	1		
název																1																		
nepracují													1				1			1														
ničím														1				1																
odvozen																			1															
počítače																				2													1	
pokračuje																					1													
praxi																						1												
procesy										1													1											
programovací												1												1										
slova										1															1									
software																										2								
stránce																											1		1					
studuje																												1					1	
technické																											1		1					
technologie										1																		1		1				
tématům															1																1			
věda																				1												1		
výpočetní																												1					1	
vztahuje																																		1

Obrázek 10: Ohodnocení výskytu jednotlivých slov RAKE algoritmem

	abstraktní	algoritmů	analýzy	daty	formálních	fungování	hardware	hlediska	informace	informační	informatika	jazyky	jelikož	jiným	konkrétnějším	název	nepracují	ničím	odvozen	počítače	pokračuje	praxi	procesy	programovací	slova	software	stránce	studuje	technické	technologie	tématům	věda	výpočetní	vztahuje
degree	3	3	3	1	2	2	2	2	2	5	1	3	2	2	2	1	2	2	1	6	1	1	2	1	2	2	2	4	2	2	2	2	2	1
frequency	1	1	1	1	1	2	1	1	2	2	1	1	1	1	1	1	1	1	1	2	1	1	1	1	1	2	1	1	1	1	1	1	1	1
degree/frequency	3	3	3	1	2	1	2	2	1	3	1	3	2	2	2	1	2	2	1	3	1	1	2	1	2	1	2	4	2	2	2	2	2	1

Obrázek 11: Ohodnocená slova jednotlivými váhami

že se nepotřebuje zaměřit na zpracovávaný jazyk a jeho syntaxi, ale zpracovává podobnost řetězců, jejichž délku si předem definujeme. Následuje vyhledání podobných zdrojů informací na webu dle klíčových slov, stažení obsahu z webových stránek, převedení do lexémů, zpracování stejným algoritmem a porovnání s již zpracovaným obsahem potencionálního souboru. Zpracovaná data se exportují do databáze systému, který umožní jejich prohlížení.

4.4.1.6 Existující systém pro správu a jeho rozšíření V současné době používaný systém nabízí dvojí přístup ke své databázi. První je určen studentům, kteří přes něj vkládají své maturitní práce v elektronické podobě. Druhý, který využívají především pedagogové, ke vkládání hodnocení jednotlivých prací at' už ze strany vedoucího práce či ze strany oponenta. Obrázek 12 znázorňuje vzhled rozhraní existujícího systému a obrázek 13 zobrazuje vazbu mezi entitamy, pro ukládání dat.

4.4.1.7 Vnitřní struktura aplikace Aplikace bude navržena modulárně, tak aby ji bylo možné dle potřeby rozšířit. Tzn. že jednotlivé dílčí části jako extrakce PDF souboru, extrakce webového obsahu, obsluha vyhledávací služby, zpracování textů pomocí slovníkuů přes analýzu textů až po export zpracovaných výsledků budou naprogramovány odděleně.

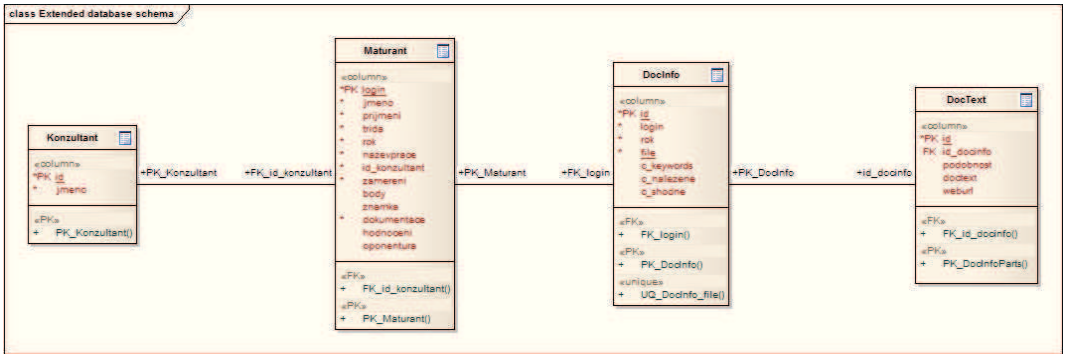
4.4.1.8 Konfigurace Samotná aplikace bude vyžadovat ke spuštění a zpracování souborů konfigurační parametry:

1. inputFolder - cesta k umístěným souborům,
2. postgreHost - adresa umístění Postgre databáze s instalovaným slovníkem,
3. postgreUser - uživatelské jméno pro přístup do Postgre databáze,
4. postgrePass - heslo pro přístup do Postgre databáze ,
5. mysqlHost - adresa umístení MySQL databáze pro uložení výstupu,
6. mysqlUser - uživatelské jméno pro přístup do MySQL databáze,
7. mysqlPass - heslo pro přístup do MySQL databáze,
8. model - cesta k souboru XML popisujícímu vstupní PDF soubor,
9. site - parametr omezující vyhledávání na jednu konkrétní doménu.

Je možné také konfigurovat nastavení jednotlivých sekcí zpracovaného vstupního souboru. Taková situace může nastat v případě, že se dokument skládá z několika kapitol a chceme zpracovávat pouze definované části. To je možné s použitím XML souboru, který popisuje vnitřní strukturu dokumentu.

Jméno	Příjmení	Třída	Mat. Rok	Název práce	Konzultant	Zaměření	Σ body	Známka	DOK	HOD	OP
Tomáš	Pechanec	P4.A	2011	Serverová utilita v Cpp	Lukáš Hapl	Info – SW	54.1	3	✗		
Štěpán	Odstrčil	E4.A	2011	Cpp Hra	Lukáš Hapl	Info – SW	95.5	1	✗		
Tomáš	Růžička	E4.B	2011	Model školy v enginu Mafie I	Lukáš Hapl	Info – SW	96.75	1	✗		
Jiří	Hajný	P4.A	2011	Web- prezentace neziskové organizace	Lukáš Hapl	Info – Multimédia	86.95	1			
Jan	Turek	P4.A	2011	Vývoj webové aplikace - Centrum Redukce Váhy	Lukáš Hapl	Info – SW	98.85	1	✗		
Vlastimil	Blinka	E4.A	2011	Cisco prvky	Lukáš Hapl	Info – SW	83.25	1	✗		
Marek	Gabriš	P4.A	2011	WWW Stranky PANO AUTODOPRAVA	Lukáš Hapl	Info – SW	87.55	1	✗		
Lukáš	Targoš	P4.A	2011	Aplikace pro i8051 v jazyku C	Lukáš Hapl	Info – SW	73.25	2	✗		
Martin	Panáček	E4.A	2011	aplikace pro smartphone	Lukáš Hapl	Info – SW	98.15	1	✗		
Radek	Chromík	P4.A	2011	CPP aplikace pro demonstraci Hammingova kodu	Lukáš Hapl	Info – SW	89.35	1	✗		

Obrázek 12: Existující systém pro správu maturitních prací

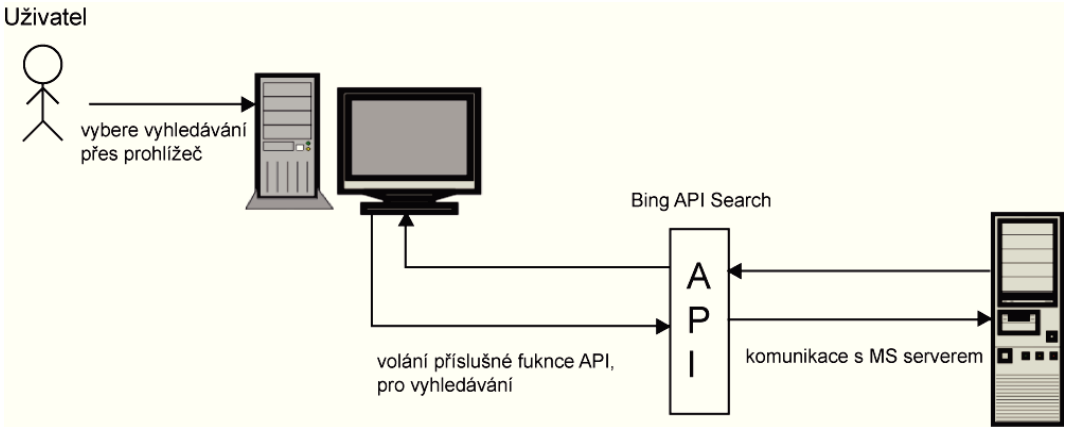


Obrázek 13: ER diagram rozšíření o entity pro ukládání zpracovaných informací

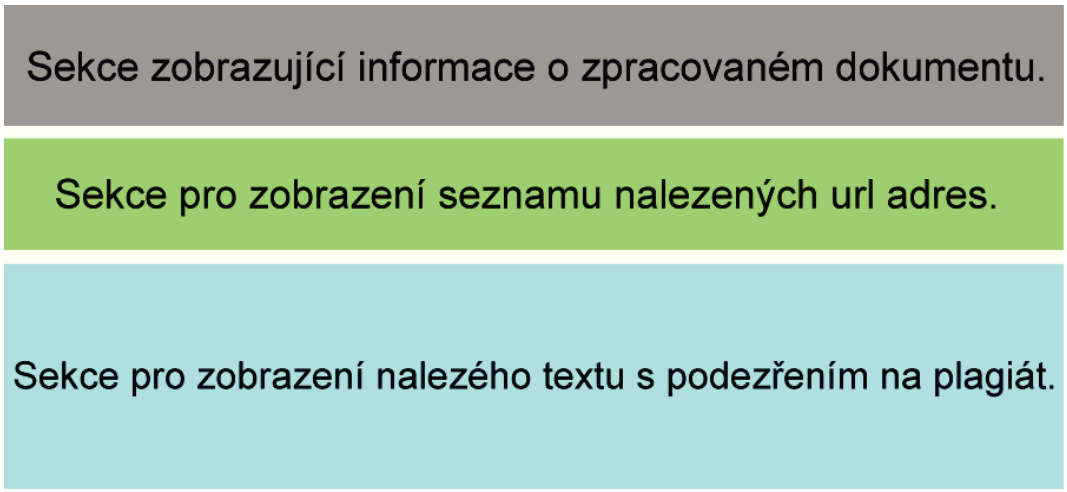
4.4.1.9 Použité metody analýzy textu Mezi použité metody analýzy a zpracování textu patří výše zmíněná metoda extrakce klíčových slov RAKE, a lokální winnowing algoritmus, které při spojení s vyhledávací službou tvoří hlavní výpočetní logiku aplikace. Nejdříve se zpracuje text, pomocí slovníků a následně se použije hashovací funkce s winnowing algoritmem. Poté se vyextrahují klíčová slova, která slouží pro vyhledání odpovídajících informací na webu. Následuje stažení webového obsahu, zpracování stejným způsobem tj. zahashování s winnowing algoritmem. Pak už se jen porovnají výsledky na pravděpodobné shody.

4.4.1.10 Komunikace s vyhledávací službou Abychom vůbec mohli porovnávat soubory, jestli jsou plagiáty nebo nejsou, musíme disponovat množinou dokumentů vůči kterým budeme porovnávání provádět. Hlavní rozdíl této práce oproti ostatním existujícím systémům je ten, že nedisponuje žádným lokálním repozitářem, ve kterém by mohly existovat předzpracované soubory, se kterými bude porovnávání probíhat. Systém využívá již zmíněné detekce klíčových slov, které potom slouží k vyhledávání podobného obsahu na internetu. Tzn. že obsah internetu je pro nás repozitářem obsahující potřebné dokumenty. Pro vyhledávání je nejjednodušší použít vyhledávacích služeb, které nabízejí giganti jako Google se stejnojmenným vyhledávacím enginem a Microsoft s vyhledávací službou Bing viz schéma na obrázku 14. v případě, že bychom chtěli využít lokálního repozitáře a mít v něm obsaženy webové stránky, museli bychom disponovat crawlerem, který potřebné stránky stáhne a bude je udržovat v aktuálním stavu, protože informace zveřejněné na webu se mohou měnit každý den, hodinu i minutu. Tím, že využijeme nalezená klíčová slova ve spojení s vyhledávací službou nepotřebujeme disponovat repozitářem, protože si vždy stáhneme aktuální informace.

Pro vyhledávání je v aplikaci použita služba Bing search, která není tak robustní jako Google search engine, ale není omezen kvótami pro počet dotazů a navíc není zpoplatněn. Oproti tomu Bing má pouze omezení, co se týče rychlosti dotazování tj. 1 dotaz za 1,5 vteřiny, ale to je pro účely této práce nevýznamné, protože aplikace bude porovnávat cca 100 dokumentů ročně.



Obrázek 14: Schématické znázornění komunikace s webovou službou



Obrázek 15: Návrh uživatelského rozhraní pro zobrazení výsledků

4.5 Uživatelské rozhraní

Návrh uživatelského rozhraní se týká pouze webové části, která bude sloužit pro zobrazování výsledků. Vzhledem k tomu, že již existuje evidenční systém, tak je potřeba jej pouze rozšířit o realtivně jednoduchou část, v podobvě webové stránky, rozdělené do tří sekcí. První sekce bude zobrazovat informace o porovnávané práci, např. autora, rok absolvování, počet nalezených klíčových slov, počet porovnaných webových stránek a počet stránek, kde byla nalezena shoda. Druhá část, bude obsahovat seznam url adres, kde se našly jednotlivé shody. Poslední sekce bude zobrazovat text vytažený z práce, který byl nalezen na příslušném webu. Obrázek 15 znázorňuje zmíněné rozdělení.

5 PLAGIWEB TOOL

Následující sekce pojednává o samotné aplikaci, která byla vytvořena na základě předchozí analýzy a návrhu. Věnuje se také instalaci podpůrných systémů, hardwarovým a softwarovým nárokům. Představuje také knihovny využité při vývoji a seznamuje uživatele s ovládáním a konfigurací. Programátorům nabízí popis jednotlivých částí systému. Produkt byl pojmenován **PlagiWeb Tool**, aby z názvu bylo jasné o jaký typ aplikace se jedná.

5.1 Instalační příručka

5.1.1 Hardwarové požadavky

Aplikace sama o sobě má poměrně velké nároky na hardware, především na pamět. Záleží na faktu, v jakém poměru je porovnáván vstupní dokument se získanými dokumenty z webu. Pokud bude aplikace provádět porovnání jednomu dokumentu vůčí tisíci dalších webových stránek, může se využití paměti vyšplhat až k 500 MB. Vše ovšem závisí na rozsahu zpracovávaných dokumentů, což se také projeví na době zpracování. Potřebný čas je přibližně 10 - 30 minut na zpracování jedné práce, tento čas ovšem započítává celý proces od načtení a předzpracování textu, přes dotazování vyhledávací služby až po následné porovnávání.

Poznámka 5.1 Systém byl vyvíjen a testován na přenosném počítači s konfigurací Intel Core 2 Duo T8100, 2,10 GHz, pamětí RAM DDR 4GB 533MHz a integrovaným grafickým čipem Intel Graphics 965 na dvou rozdílným operačních systémech: Microsoft Windows 7 64bit, a linuxovém operačním systému Fedora 16. Rychlost aplikace je také závislá na tom, v jaké míře je využívána pamět', procesor a internetové připojení ostatními aplikacemi a procesy.

5.1.2 Softwarové požadavky

Hlavním softwarovým požadavkem pro běh systému je zprovoznění Java Virtual Machine, Postgre SQL databáze s instalovanými slovníky cspell, a MySQL databází pro uložení výstupních dat. Vývoj probíhal ve vývojovém prostředí Netbeans IDE 7.1 s instalovanou Java verzí 1.7.0_01, Postgre SQL databází verze 8.3.13 a MySQL serveru 5.5, takže bych pro spuštění doporučoval stejné verze případně vyšší. Ovšem není potřeba instalovat zmíněné databázové systémy, pokud využijete již existujících umístěných kdekoliv v síti, musíte pouze nastavit potřebné parametry v konfiguračním souboru.

5.1.3 Instalace

Instalace Postgre SQL databáze nepatří mezi složité, pokud použijeme verzi s instalátorem, který nás jednoduše provede celým procesem, doporučuji přímo při instalaci

vybrat navíc balík phpPgAdmin, což je webové rozhraní pro práci s databází. Instalace probíhá pod uživatelem postgres, což je jediný uživatel, který je k dispozici k inicializaci databáze. Tento uživatel je svým způsobem administrátorský uživatel postgre databáze, protože díky němu můžeme přidávat i další uživatele, měnit oprávnění a další. Nyní se zaměříme na import použitých cspell slovníků, které jsou nezbytné pro vyvíjenou aplikaci. Instalační balík pro databáze 8.3 a vyšší je možné stáhnout z http://www.pgsql.cz/data/czech.tar.gz a lze jej aplikovat pouze na databáze s kódováním UTF8. Balík obsahuje soubory czech.dict, czech.affix, czech.stop, které musí být rozbaleny a přesunuty do adresáře s postgre, konkrétně do share/tsearch_data, cesta závisí na použitém operačním systému a verzi instalace [31]. Poté, co jsou všechny soubory zkopírovány, můžeme spustit SQL zobrazený ve výpise 1.

```
CREATE TEXT SEARCH DICTIONARY cspell
(template=ispell, dictfile = czech, afffile =czech, stopwords=czech);
CREATE TEXT SEARCH CONFIGURATION cs (copy=english);
ALTER TEXT SEARCH CONFIGURATION cs
ALTER MAPPING FOR word, asciiword WITH cspell, simple;
```

Výpis 1: SQL pro instalaci slovníku do postgre databáze

Dotaz z výpisu 2 je nutné spustit na správné databázi, kterou používáme v aplikaci. Pokud se dotaz provedl v pořádku, tak výstup z funkce pro dotaz na řetězec *„Plagiátorstvím se rozumí opisování, přebírání a publikování cizích myšlenek či výsledků výzkumu a jejich vydávání za své bez uvedení původního zdroje“* by vypadal jako v tabulce 6.

```
SELECT * FROM TS\_DEBUG('cs','Plagiátorstvím se rozumí opisování, přebírání a
    publikování cizích myšlenek.');
```

Výpis 2: SQL pro ověření funkce fulltextového slovníku

Pro využití Postgre SQL databáze v programovacím jazyce Java musíme disponovat JDBC ovladačem, který zprostředkovává komunikaci s databázovým serverem. Jedná se o malou jar knihovnu, která se dá naimportovat do jakékoliv Java aplikace a je možné ji okamžitě použít.

Následuje instalace MySQL serveru, která probíhá obdobně. Nejdříve stáhneme požadovaný instalátor dle verze našeho systému (64bitový nebo 32bitový) a poté jej spustíme. Pro spuštění musíme disponovat administrátorskými právy a dostatkem volného místa jinak se instalace nezdaří. Pro rozbalení, spuštění a vytváření databází je minimum stanoveno na 200MB. Instalační průvodce vás provede všemi kroky, od toho kam databázi instalovat, kolik k ní bude přistupovat uživatelů přes přístupové heslo root uživatele až k tomu o jaký typ stroje se jedná, jestli o developerský nebo server. Systém Windows nám umožní MySQL server spouštět jako službu, kvůli tomu jsou potřeba zmíněná práva. Opět doporučuji doinstalovat balíček phpMyAdmin, které slouží jako webové rozhraní k obsluze MySQL databází. Jeho obsluha je jednoduchá a intuitivní. Pro běžné použití nám stačí znalost samotného SQL a toto webové rozhraní nám jen usnadní práci [32]. Pro obě databáze je společným instalačním prvkem nutná podpora TCP/IP protokolu.

alias	description	token	dictionaries	dictionary	lexemes
word	Word, all letters	Plagiátorstvím	{cspell,simple}	cspell	{plagiátorství}
blank	Space symbols		{}	NULL	NULL
ascii	Word, all ASCII	se	{cspell,simple}	cspell	{}
blank	Space symbols		{}	NULL	NULL
word	Word, all letters	rozumí	cspell,simple	cspell	rozumět
blank	Space symbols		{}	NULL	NULL
word	Word, all letters	opisování	cspell,simple	cspell	opisování
blank	Space symbols		{}	NULL	NULL
word	Word, all letters	přebírání	cspell,simple	cspell	přebírání
blank	Space symbols		{}	NULL	NULL
ascii	Word, all ASCII	a	cspell,simple	cspell	
blank	Space symbols		{}	NULL	NULL
word	Word, all letters	publikování	cspell,simple	cspell	publikování
blank	Space symbols		{}	NULL	NULL
word	Word, all letters	cizích	cspell,simple	cspell	cizí
blank	Space symbols		{}	NULL	NULL
word	Word, all letters	myšlenek	cspell,simple	cspell	myšlenka
blank	Space symbols		{}	NULL	NULL

Tabulka 6: Výsledek dotazu na cspell slovník

Pro spuštění je potřeba mít nainstalovaný odpovídající balík JRE případně JDK (pokud budeme chtít aplikaci dále rozvíjet). Stáhneme odpovídající balík a spustíme instalaci. Opět budeme provedeni několika kroky, od toho kam se námi vybraný balík nainstaluje, jestli chceme nastavit PATH, což je proměnné prostředí operačního systému [10] [11] [33].

Všechny instalační balíky a knihovny použité při vývoji jsou přiloženy na CD k samotné práci.

5.2 Uživatelská příručka

V této kapitole se podíváme na ovládání samotné aplikace pro detekci, ukážeme si jak správně nakonfigurovat parametry pro úspěšné spuštění a popíšeme si význam jednotlivých prvků ve vstupním XML souboru. Aplikaci je možné spustit přiloženým run.bat souborem v případě, že používáte operační systém Windows, nebo run.sh souborem v případě že disponujete operačním systémeme na bázi linuxu. Výpis 3 zňázorňuje jak vypadá obsah bat souboru, ve kterém je také zapsána konfugirace pro spuštění. Program lze spustit z příkazové řádky. Popis jednotlivých parametrů je uveden v kapitole 4.4.1.8.

```
@ECHO OFF
REM Input folder path (without file extension)
   set INPUT_FOLDER="c:/Enviroment/workspace/Plagi/iofiles/"

REM Postgre host
   set POSTGRE\_HOST="localhost:5432/plagi"

REM Postgre username
   set POSTGRE\_USER="plagi"

REM Postgre password
   set POSTGRE\_PASS="plagitest"

REM MySQL host
   set MYSQL\_HOST="localhost"

REM MySQL username
   set MYSQL\_USER="root"

REM MySQL password
   set MYSQL\_PASS="admin"

REM Model
   set MODEL="notused"

REM Site
   set SITE="notused"

REM run it

java -cp plagi.Plagi inputFolder="%INPUT_FOLDER%" postgreHost="%POSTGRE_HOST%"
     postgreUser="%POSTGRE_USER%" postgrePass="%POSTGRE_PASS%" mysqlHost="%
     MYSQL_HOST%" mysqlUser="%MYSQL_USER%" mysqlPass="%MYSQL_PASS%" model="%
     MODEL%" site="%SITE%"
```

Výpis 3: Obsah souboru run.bat

Soubor XML, jak již bylo zmíněno slouží, k popisu zpracovávaného PDF souboru, ve kterém můžeme nadefinovat, které části chceme porovnávat a které ne. Do budoucna se má toto nastavení dále rožšiřovat např. o ruční nastavení významnosti jednotlivých sekcí. z toho důdu byl pro popis vybrán XML soubor, který aplikace umí zpracovat a je možné ji tak rozšířit o další funkce. Důležitými paramtery jsou *name*, které jsou vnořeny do *chapter* a popisují názvy jednotlivých sekcí oddělujících dokument. Parametr *process* označuje, jestli má být sekce zpracována, či nikoliv. Pro tento účel zde slouží dvě hodnoty, první y - což znamená ano a sekce bude zpracována, druhý n - což znamená ne a sekce nebude zpracována. Výpis 4 zobrazuje obsah XML souboru.

```
<document>
  <chapters>
    <chapter>
      <name>abstrakt</name>
      <process>y</process>
    </chapter>
    <chapter>
      <name>obsah</name>
      <process>n</process>
    </chapter>
    <chapter>
      <name>úvod</name>
      <process>y</process>
    </chapter>
    <chapter>
      <name>1. cíle práce</name>
      <process>y</process>
    </chapter>
    <chapter>
      <name>2. výběr technologií pro řešení</name>
      <process>y</process>
    </chapter>
    <chapter>
      <name>3. způsoby řešení a použité postupy</name>
      <process>y</process>
    </chapter>
    <chapter>
      <name>4. zhodnocení dosažených výsledků</name>
      <process>y</process>
    </chapter>
    <chapter>
      <name>shrnutí</name>
      <process>y</process>
    </chapter>
  </chapters>
</document>
```

Výpis 4: Výpis souboru model.xml popisující strukturu PDF dokumentu

Operační systém	Microsoft Windows 7 Professional a Fedora 16
Vývojové prostředí (IDE)	NetBeans IDE 7.1
Verze Java	JDK 1.7.0_01
Databáze	Postgre SQL 8.3.13-1, MySQL 5.5.23
Webový server	WampServer 2.2D
Vyhledávací služba	Bing API Search 2.0
Prohlížeč	Google Chrome 18.0.1025.162

Tabulka 7: Seznam použitých technologií při vývoji

Název	**Popis**
Postgre SQL 9.1 JDBC driver	JDBC ovladač sloužící k napojení aplikace na Postgre databázi
MySQL Connector 5.0.8	JDBC ovladač k připojení aplikace na MySQL databázi
Apache PDFBox 1.7.0	Balíček umožňující práci s PDF soubory
Jsoup 1.6.2	Java HTML parser umožňující práci s HTML obsahem webových stránek
Vyhledávací služba	Bing API Search 2.0
Prohlížeč	Google Chrome 18.0.1025.162

Tabulka 8: Seznam použitých technologií při vývoji

V závislosti na počtu porovnávaných prací, se po skončení běhu každého zpracovaného PDF souboru, exportují data do výstupní MySQL databáze.

5.3 Programátorská příručka

Tato kapitola práce poskytuje základní pohled na použité principy a technologie z programátorského hlediska. Je tedy takovou základní dokumentací, popisující vnitřní strukturu aplikace a má sloužit programátorům, aby se rychleji dovedli zorientovat v samotném kódu.

5.3.1 Použité technologie

Tabulka 7 uvádí verze všech použitých technologií a nástrojů, které byly použity při vývoji PlagiWeb Tool.

Seznam použitých knihovem, které sloužily jako podpůrné části, je uveden v tabulce 8 včetně jejich základního popisu a využití v aplikaci.

Název	Popis
AppId	Application ID, umožňující nám zavolat službu Bing API
Query	Slouží k sestavení dotazu z námi extrahovaného klíčového slova plus domény, na kterou se hledání vztahuje
Sources	Určuje v jakých zdrojích se bude vyhledávat
Web.Count	Určuje kolik výsledků hledání budeme chtít od služby vracet na jeden dotaz
Web.FileType	Definuje typ hledaných dokumentů

Tabulka 9: Popis proměnných nezbytných pro sestavení dotazu

5.3.2 Bing API

Jak už bylo řečeno Bing API verze 2, umožňuje vývojářům dotazovat se vyhledávácí služby, z programového kódu na Bing Engine a získávat tak výsledky hledání, které se dají dále zpracovávat. Lze jej využít pro získání informací z Internetu (nejen v textové podobě, umí také vyhledávat obrázky, novinky, instantní odpovědi a videa), zlepšuje vyhledávání a zvyšuje možnost získat relevantní výsledky (v případě použití cíleného hledání na určených doménách), umožňuje najít, kde se daná informace nachází a také umí překládat pojmy a bloky textu [30]. k využití API je potřeba vytvořit si účet na službě Windows Live a registrovat svoji aplikaci. Poté co splníme tyto dva kroky získáme uníkátní Application ID, které nám umožní využívat rozhraní přímo v programu. Nejdůležitějším prvkem, je složení dotazu pro vyhledávací službu, která vrací výsledky v podobě XML dat. Výpis 5 ukazuje hlavní metodu pro sestavení dotazu, která je klíčovým prvkem. Jednotlivé prvky a proměnné jsou popsány v tabulce 9. Ostatní metody použité pro parsování jsou také sice nezbytné, ale pro nás je nejpodstatnější ukázka sestavení dotazu.

```
private static String BuildRequest(String query) {
        String AppId = "39916A197E3CD3969ACC3DA7077E9FFE4A597B9A";
        String requestString = " http: // api . search . live . net/xml.aspx?"
                + "AppId=" + AppId
                + "&Query=" + Settings.getSite() + "\"" + query + "\""
                + "&Sources=Web"
                + "&Version=2.0"
                + "&Web.Count=50"
                + "&Web.Offset=0"
                + "&Web.FileType=HTML"
                + "&Web.Options=DisableHostCollapsing+DisableQueryAlterations";

        return requestString;
    }
```

Výpis 5: Metoda sestavující dotaz na Bing API

5.3.3 Popis tříd

Aplikace je tvořena několika hlavními třídami, které obstarávají zpracování vstupních PDF souborů, použití detekce klíčových slov, redukce a zahashování textů v podobě winnowing algoritmu, následné porovnání a výpočet výsledků. Základní komponenety jsou uvedeny v tabulce 10 se stručným popisem, k čemu jsou určeny.

Pokud by tedy v budoucnu nastala potřeba rozšířit podporu o vstupní formát, přidala by se např. třída DOCextractor, která by zajišťovala zpracování „doc“ dokumentů. A upravilo by se pouze volání extraktní třídy, podle použité přípony. Popis dokumentů v podobě XML modelu by zůstal stejný. Obdobné je to v případě použití dalších statistických metod, které by se zakomponovaly do třídy MyContent a zanořily se do třídy ContentPart, aby měly přístup k odpovídajícím obsahům jednotlivých doikumentů. Tak by zůstala zachávana současná integrita a nové metody by se mohly volat jako přídavné, případně by mohly nahradit již použité metody.

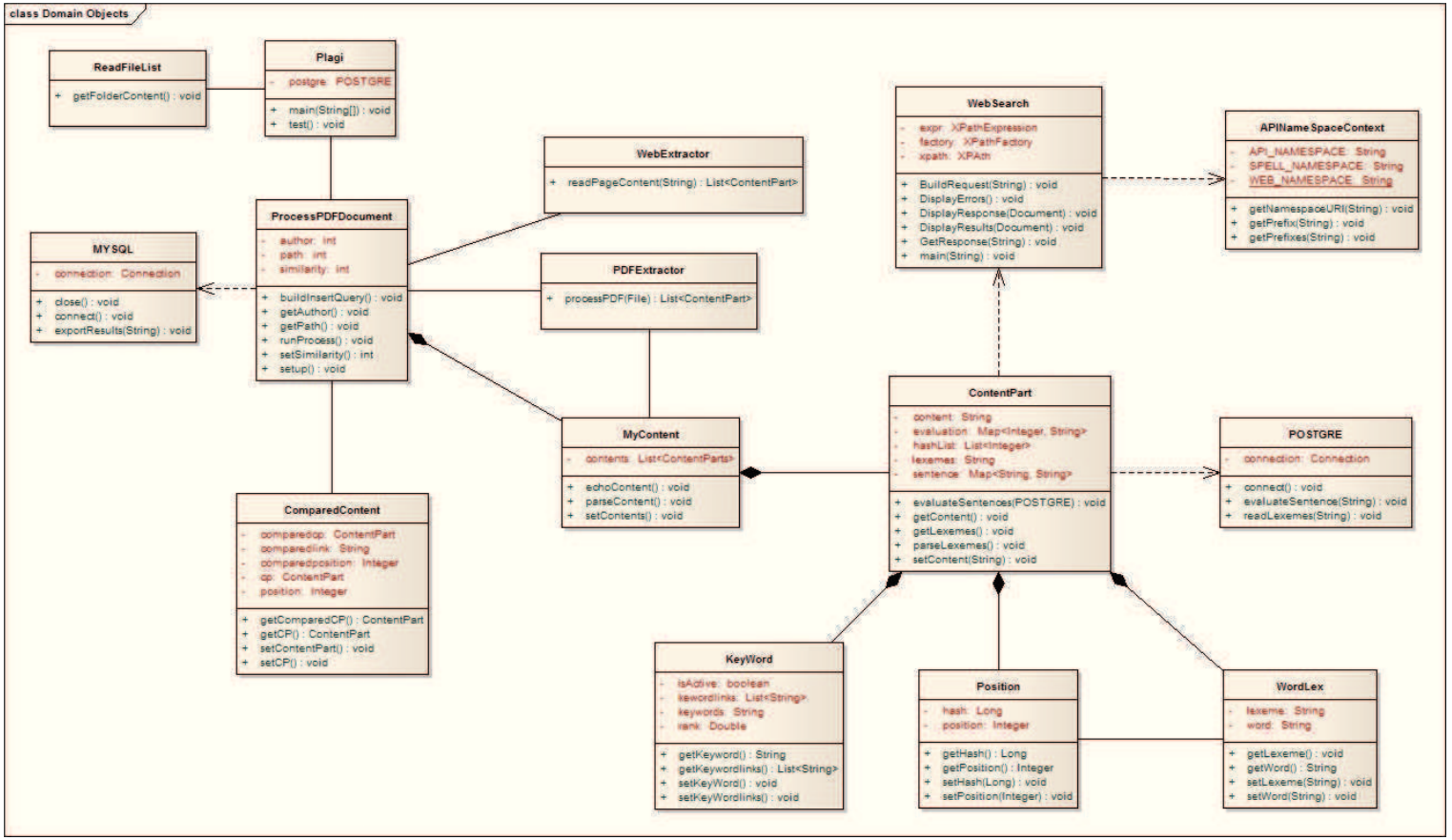
5.4 Rozšíření do budoucna

Na samotný závěr této kapitoly následuje seznam vlastností a funkcí, které ještě více vylepší, zrychlí a zefektivní systém PlagiWeb Tool. Pomůžou zdokonalit jeho ovládání a co nejvíce zautomatizují celý process odhalování plagiátů:

- kontrola gramatiky,
- rozšíření o další statistické metody,
- rozšíření o zpracování nejen závěrečných prací,
- rozšíření o vlastní repozitář pro rychlejší zpracování,
- snížení paměťové náročnosti,
- umožnit přímé spojení na ftp adresář pro zpracování dokumentů,
- zvážit rozšíření o jinou metodu využívající n-gramy a repozitář n-gramu,
- použít Google API Search,
- využít YQL,
- umožnit editovat nalezené výsledky,
- integrovat aplikaci do systému školy a zpřístupnit ji v podobě služby.

Název třídy	**Popis**
ProcessPDFDocument	Účelem třídy je zajistit zpracování jednoho PDF dokument. Tzn. zajišťuje překlad dokumentu na text, detekci klíčových slov, následnou lexemizaci, zahashování, zjišťuje přes rozhraní webové služby Bing relevantní odkazy, stahuje jejich obsah a zpracováváho obdobným způsobem. Následně provede porovnání, zpracování a export výsledků.
MyContent	Představuje jeden konkrétní dokument (PDF nebo webový) a jeho účelem je zpracovat jeho části, které jsou uloženy v ContentPart.
ContentPart	Tato třída sama o sobě provádí veškerou nezbytnou práci, co se týče zrpacování textu. Převod do lexémů, hledání klíčových slov, zpracování hashů, a porovnávání s jinými částmi především webových dokumentů.
PDFExtractor	Má na starosti načtení příslušných PDF souborů a jejich převod na text.
WebSearch	Zajišťuje vyhledávání odpovídajících webových dokumentů pomocí klíčových slov.
WebExtractor	Obdobná třída jako PDFExtractor, ale zajišťuje extrakci webového obsahu.
WordLex	Jedná se o objekt, který obsahuje slovo a jeho odpovídající interpretaci v podobě lexému.
Position	Objekt, který obsahuje hash odpovídající části textu a jeho pozici, kde se nachází.
KeyWord	Objekt obsahující klíčové slovo a seznam odpovídajících odkazů, které byly podle něj nalezeny.
POSTGRE	Zajišťuje komunikaci s Postgre SQL databází.
MYSQL	Zajišťuje komunikaci s MySQL databází.

Tabulka 10: Seznam tříd a jejich popis



Obrázek 16: Třídní diagram programu PlagiWeb Tool

6 VÝSLEDKY EXPERIMENTŮ

Poslední kapitola práce je rozdělena do tří částí a věnuje se porovnávání metod a principů, které byly použity při vývoji a odůvodňuje, proč byly vybrány. První je zaměřena na analýzu a zpracování textů a jejich přípravu na porovnávání s ostatními dokumenty. Druhá nabízí pohled na odlišné metody pro detekci klíčových slov. A poslední z nich se zabývá samotným srovnáním výsledků navržené aplikace s několika zahraničními systémy uvedenými v kapitole 2.4.1.

6.1 Porovnávání textů

Následující sekce se zabývá metodami zpracováním do podoby vhodné k porovnávání. Takže se svým způsobem jedná o typy indexování textů, které jsou v dnešní době nezbytné pro jejich rychlé zpracování.

6.1.1 MOZAIKA

Ve skriptech [18] se název vysvětluje jako zkratka pro „*na morfologickém odvozování založené automatické indexování koherentními agregáty*“ a označuje metodu automatického indexování odborných textů psaných v češtině nebo v podobných jazycích, která byla vyvinuta skupinou matematiků na MFF UK v Praze. Jejím cílem je přiřadit vstupnímu textu selekční obraz vytvořený podle čtyř základních zásad.

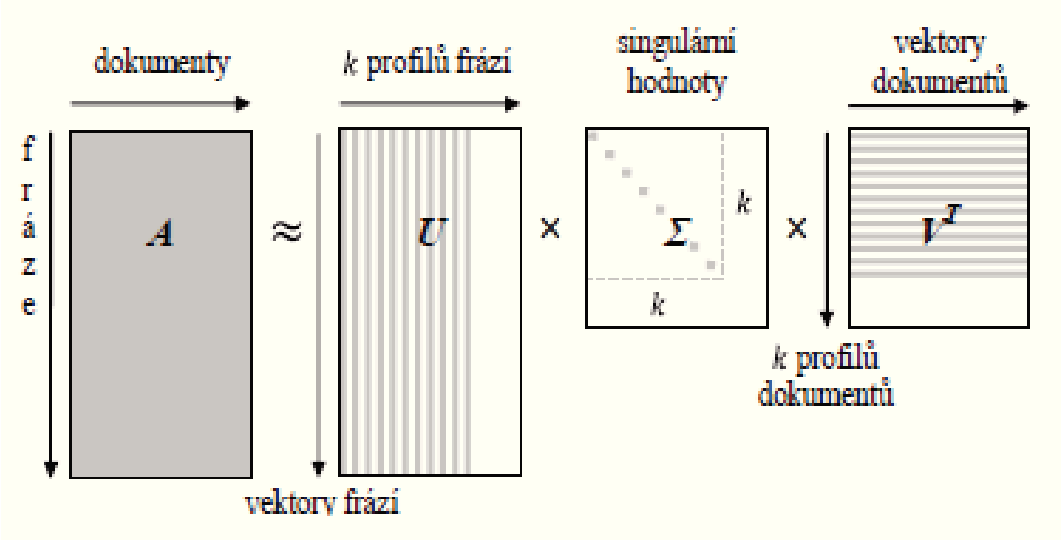
1. Selekční znaky se vybírají z textu.
2. Žádoucí jsou termíny a jména maximálně charakterizující obsah textu
3. Upřednostňují se víceslovné výrazy.
4. Nepoužívá se žádný předem připravený slovník termínů.

Vstupem je libovolný text, nejlépe však celý dokument v původním stavu s přesně zachovanými edičními znaky (formátováním nadpisů, odstavců, poznámek, abstraktu apod.). Je to důležité proto, aby bylo možné jednoznačně oddělit dokument.

Výstupem je seznam terminologických výrazů z textu nejcharakterističtějších pro jeho obsah, seřazených podle váhy v daném textu včetně jeho pozic, kde se vyskytují. Tato metoda je ideálním kandidátem pro zpracování dokumentů určených pro uložení do lokálních repozitářů, především tedy pro uložení publikovaných knih, článku apod., ne pouze pro zpracování informací z libovolných webových stránek, které často nemají potřebnou strukturu (i v případech kdy se jedná o oficiální zdroje).

6.1.2 LSA - latentní sémantická analýza

Předpokladem pro použití této metody, je že disponujeme předzpracovaným textem, u kterého jsou odstraněny stop-slova, zbylá slova jsou převedeny do kořenových tvarů a



Obrázek 17: Dekompozice textů do matic frází s využitím SVD

jsou extrahovány fáze. Tj. nejdříve se text ošetří o nadbytečné prvky, které nemají při zpracování význam a poté se vytváří seznam frází, které popisují daný dokument. Následně se využíva singulární dekompozice využívané právě v LSA, kde informace o zpracovaném textu představují matici, která je převedena na singulární hodnoty. Ty poté slouží k přímému porovnávání s jinými dokumenty (které musí být nejdříve převedeny) viz obrázek 17. LSA metoda je ideální kandidát na využití n-gramů, které by představovaly vektorový prostor jednotlivých dokumentů a usnadnily by tak porovnávání [19].

6.1.3 Lokální winnowing

Postup, který využívá přímo principu n-gramů a definuje jak mají vznikat a jak se s nimi má pracovat. v první řadě určuje, že se text má rozdělit do n-gramů např. s rozkladem po sedmi slovech (která se zahashují) a ze kterých pak vznikají bloky 4-gramů, které se vzájemně překrývají. Dále pak definuje, jak redukovat totu množinu o právě překrývající se části. v příkladu na obrázku 18 je uvedena ukázka jak algoritmus funguje v případě, když využívá tvorby n-gramů s posunem o 1 znak. Tato metoda je mnohem jednodušší a výkonově méně naročnější na použití a navíc není závislá na žádne znalosti zpracovávaného jazyka oproti LSA [5] [2] [1].

6.2 Detekce klíčových slov

Ukázalo se, že pro úspěšné a efektivní porovnávání textů je nezbytné disponovat vhodnou technikou umožňující indexování. Jenže pokud nedisponujeme množinou dokumentů k porovnávání a chceme využít vyhledávacích služeb, pro hledání dokumentů s podobným obsahem, tak musíme vybrat vhodnou metodu extrakce klíčových slov (jedná se hlavně o případ, když dokument není popsán klíčovými slovy). Existuje několik metod,

Vstupní text:

Cílem není chránit informace

Předzpracovaný text:

cilemnenichranitinformace

Převod do n-gramů délky 5:

cilem ilemn lemne emnen mneni nenic enich nichr ichra chran hrani ranit aniti nitif tinfo infor nform forma ormac rmace

Hypotetický převod do hashů:

77, 75, 45, 18, 98, 50, 63, 20, 12, 65, 78, 50, 17, 88, 19, 67, 39, 42, 8, 26

Převod do skupin n-gramů po 4 ve skupině:

(77, 75, 45, **18**)	(75, 45, 18, 98)	(45, 18, 98, 50)	(18, 98, 50, 63)
(98, 50, 63, 20)	(50, 63, 20, **12**)	(63, 20, 12, 65)	(20, 12, 65, 78)
(12, 65, 78, 50)	(65, 78, 50, **17**)	(78, 50, 17, 88)	(50, 17, 88, 19)
(17, 88, 19, 67)	(88, 19, 67, 39)	(**19**, 67, 39, 42)	(67, 39, 42, **8**)
(39, 42, 8, 26)			

Winnowingem vybrané n-gramy:

18, 12, 17, 19, 8

Pozice vybraných n-gramů v textu:

[18, 3] [12, 8] [17, 12][19, 14][8, 18]

Obrázek 18: Použití lokálního winnowingu s redukcí frází

které se dají aplikovat na vyhledávání, založené jak na slovnících, tak na sémantických analýzách textu. V mém případě jsem potřeboval využít metody, která je jazykově nezávislá, protože v další části jsem porovnával aplikaci s jinými systémy, které pracují především s anglickým jazykem. Mezi principy, které jsem nastudoval patří Carpenova metoda, TextRank, NP chunks a RAKE metoda, která se nakonec jevila jako nejvýhodnější, nejefektivnější a také nejrychlejší. Výsledky testů a porovnání uvedených metod (kromě Carpenovy metody, kterou jsem neimplementoval) můžete vidět v tabulce 11.

6.2.1 TextRank

Princip metody spočívá v tom, že používá syntaktické filtry pro identifikaci tzv. POS značek (part-of-speech - část řeči), které vybraná slova ohodnocují jako klíčová nebo ne. Vybraná slova se poté akumulují v podobě grafu, který popisuje jejich výskyt a do kterého se zaznamenává hodnocení slov na základě jejich sdružení s ostatními slovy. Poté jsou z tohoto žebříčku jsou vybrána pouze top slova, vybraná klíčová slova, která jsou přilehlá, případně se v jejich blízkosti vyskytují jiná slova, se spojují do tzv. multi-klíčových slov. Zjištěná slova poté popisují dokument, a lze je použít pro vyhledávání. Přesnost detekce se uvádí cca 70% [1].

6.2.2 Carpenova metoda

Základní myšlenka pro posuzování dokumentů vůči hledaným slovům, je tak že si uděláme statistiku výskytů jednotlivých slov. Čím větší frekvence jednotlivých slov, tím je i vyšší váha a tím je více pravděpodobné, že dokument je zaměřen právě na toto téma. Jenže Carpenova metoda vynechává úplně tento typ statistiky. Vychází z toho, že klíčová slova charakterizující dokument, se získají ne podle frekvence, ale podle toho jak jsou řazena za sebou. Důležitá slova se shlukují dohromady a ty méně důležitá se se náhodně vyskytují v dokumentu. Z tohoto pohledu je to i logické, protože autor se zabývá vysvětlováním jednoho tématu a je velmi pravděpodobné, že v dalších částech textu se jim už věnovat nemusí [27].

6.2.3 NP chunks s využitím tagů

Tato metoda využívá rozdělování vět do nepřekrývajících se částí tzv. NP chunking, kdy jednotlivé kusy představují jmenné fráze. Podstatnou roli zde hraje také využití POS značek, které identifikují například slovesa daného jazyka. Tzn. že metoda umožňuje zaměřit se na konkrétní části a skupiny slov, protože se využívá sémantické analýzy a vzorů analyzovaného jazyka. Tento způsob lze využít například s existující aplikací Apache OpenNLP, která má integrovanou logiku NP chunks. Má pouze jednu nevýhodu a to, že je zaměřena jazykově a podporuje pouze několik cizích jazyků a pro použití češtiny by bylo třeba nadefinovat slovník a správně ji naučit na testovacích vzorcích [1] [2].

Metoda	Extrahovaná slova	Střední hodnota	Správná slova	Střední hodnota	Přesnost
RAKE	6052	12.1	2037	4.1	33.7
TextRank	6784	13.6	2116	4.2	31.2
NP chunks	7815	15.6	1973	3.9	25.2

Tabulka 11: Porovnání metod pro extrakci klíčových slov

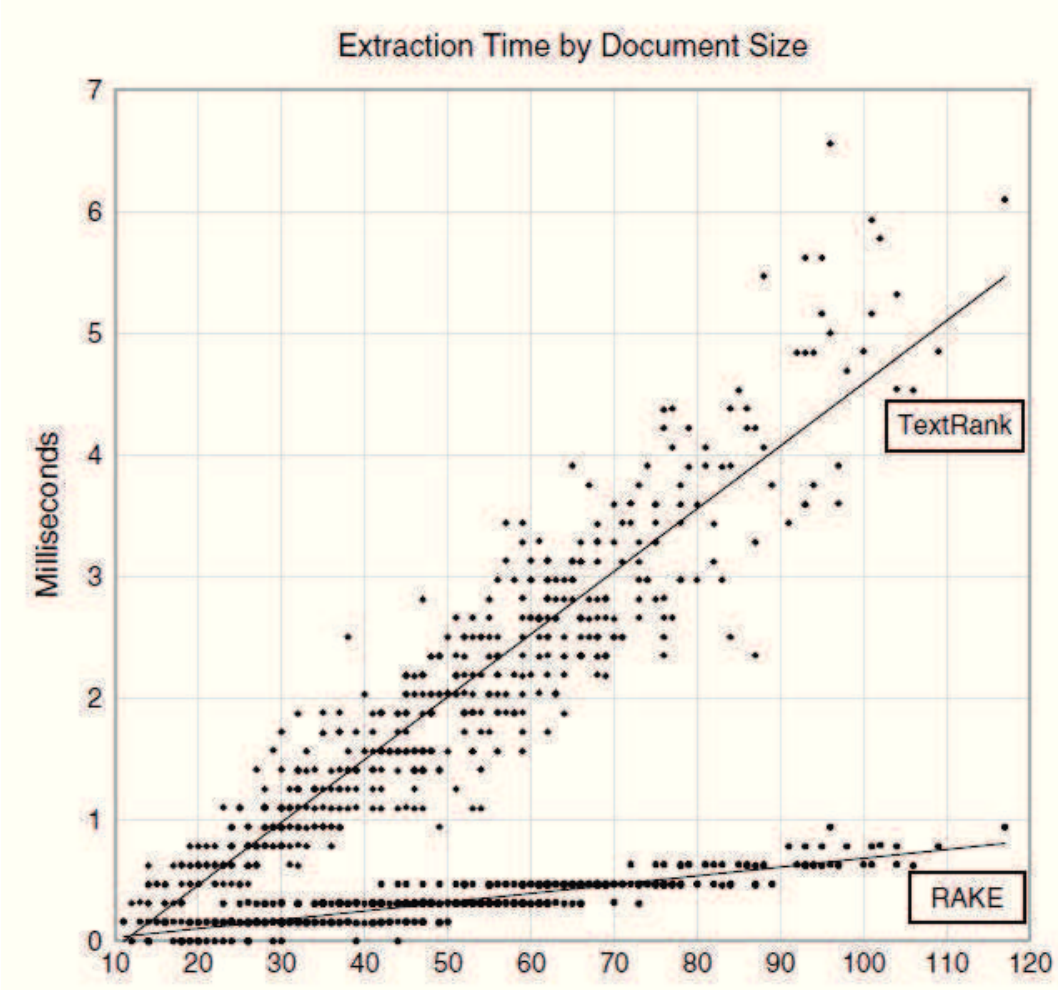
6.2.4 RAKE

RAKE metodou jsme zabývali detailněji části 4.4.1.4, takže si ji alespoň trošku přiblížíme. Ve své podstatě kombinuje TextRank metodu a Carpenovu metodu. Za prvé využívá statistického výpočtu četnostli jednotlivých slov a za druhé využívá toho, jak jsou jednotlivá slova řazena vedle sebe. Velkou výhodou je, že využívá slovníku testovaného jazyka a jeho definovaných stop slov, což nemusejí být pouze interpunkční znaménka, spojky a zájmena, ale mohou být definovány uživatelem. Navíc uživatel může nadefinovat svůj slovník, který bude zvyšovat hodnotu slov. Tohoto principu se využívá, u prací, které jsou jednosměrně zaměřeny a spadají do stejné kategorie. U prací z širšího spektra oborů tento slovník použit nelze, jedině v případě, že by jsme u každého textu definovali o jakou kategorii se jedná. Přesnost metody je cca 70%. Důvod, proč jsem vybral implementaci této metody je jasný, za prvé není jazykově závislá a nepotřebuje tak slovník s popisem významů jednotlivých slov a za druhé jednoduše kombinuje princip TextRanku a Carpenovy metody [1].

Údaje v tabulce 11 zobrazují výsledky porovnání zmíněných metod (bez Carpenovy metody, která se využívá přímo v RAKE metodě), od celkového počtu extrahovaných slov, přes počet výsledných slov až po přesnost. Testování bylo provedeno na množině 100 dokumentů s obsahem 5000 slov na jeden dokument. Sloupec *Extrahovaná slova* říká, kolik bylo celekem zjištěno klíčových slov, sloupec *Správná slova* kolik slov bylo vybráno jednotlivými metodami jako správný vzorek, použitelný pro popis dokumentů.

6.3 Srovnání s aplikacemi

Subkapitola je věnována porovnání výsledků vyvinuté aplikace s dvěmi aplikacemi, které využívají vlastních metod pro vyhledávání plagiátů. Obě využívají svého repozitáře, ve kterém mají uloženy informace z webu a v případě nalezení shody zobrazují nalezený odkaz s procentuálním vyjádřením podobnosti. Test všech aplikací byl prováděn na dokumentu psaném v anglickém jazyce, protože srovnávané programy neuměly rozeznávat češtinu, narozdíl od vyvinuté aplikace, která je jazykově nezávislá (záleží pouze na fulltextovém slovníku instalovaném v Postgre SQL databází a přepnutí nastavení z českého jazyka na anglický). Mezi srovnávané aplikace patří Viper a Plagiarism-Detector, první z nich vyžaduje registraci, druhá má omezen počet použití, po jejímž vyčerpání je uživatel vyzván, aby si produkt koupil.



Obrázek 19: Porovnání rychlosti metody TextRank a RAKE

6.3.1 Testovací dokument

Dokument sloužící pro otestování aplikací, byl složen ze čtyř částí sestavených z různých témat. Patřila mezi část týkající se informačních a komunikačních technologií (zdroj: http://www.flinders.edu.au/science_engineering/csem/disciplines/itse/), spalovacích motorů (zdroj: http://en.wikipedia.org/wiki/Engine), automobilů a výtažku z učebnice socilogie.

Příklad 6.1
Část textu testovaného dokumentu:

„Information and communications technology or information and communication technology, usually abbreviated as ICT, is often used as an extended synonym for information technology (IT), but is usually a more general term that stresses the role of unified communications and the integration of telecommunications (telephone lines and wireless signals), computers, middleware as well as necessary software, storage and audiovisual systems, which enable users to create, access, store, transmit, and manipulate information. In other words, ICT consists of IT as well as telecommunication, broadcast media, all types of audio and video processing and transmission and network based control and monitoring functions. The expression was first used in 1997 in a report by Dennis Stevenson to the UK government and promoted by the new National Curriculum documents for the UK in 2000."

■

6.3.2 Viper - Scan My Essay

Aplikace Viper(Scan My Essay) jako první testovaná v testu neobstála, jako jediná nenašla žádnou shodu i přesto, že vykazovala nějakou činnost a nejvíce času jí zabralo připojování do repozitáře a čekání na přidělení přístupu (tato doba není připočtena k době trvání). Nástroj je možné stáhnout z webových stránek http://www.scanmyessay.com/.

6.3.3 PlagiWeb Tool

PlagiWeb Tool (nástroj, který byl předmětem této práce) v testu nedopadl nejhůře, ale v porovnání s Plagiarism-Detectorem, který značně převyšuje jeho možnosti trochu zaostával. Slabší výsledky se dají oddůvodnit tím, že aplikace, je jazykově nezávislá a využívá jenom slovníků daného jazyka bez jakéhokoliv popisu. Takže mohlo dojít k určité chybě způsobené právě použitým slovníkem a použitými stop slovy.

6.3.4 Plagiarism-Detector

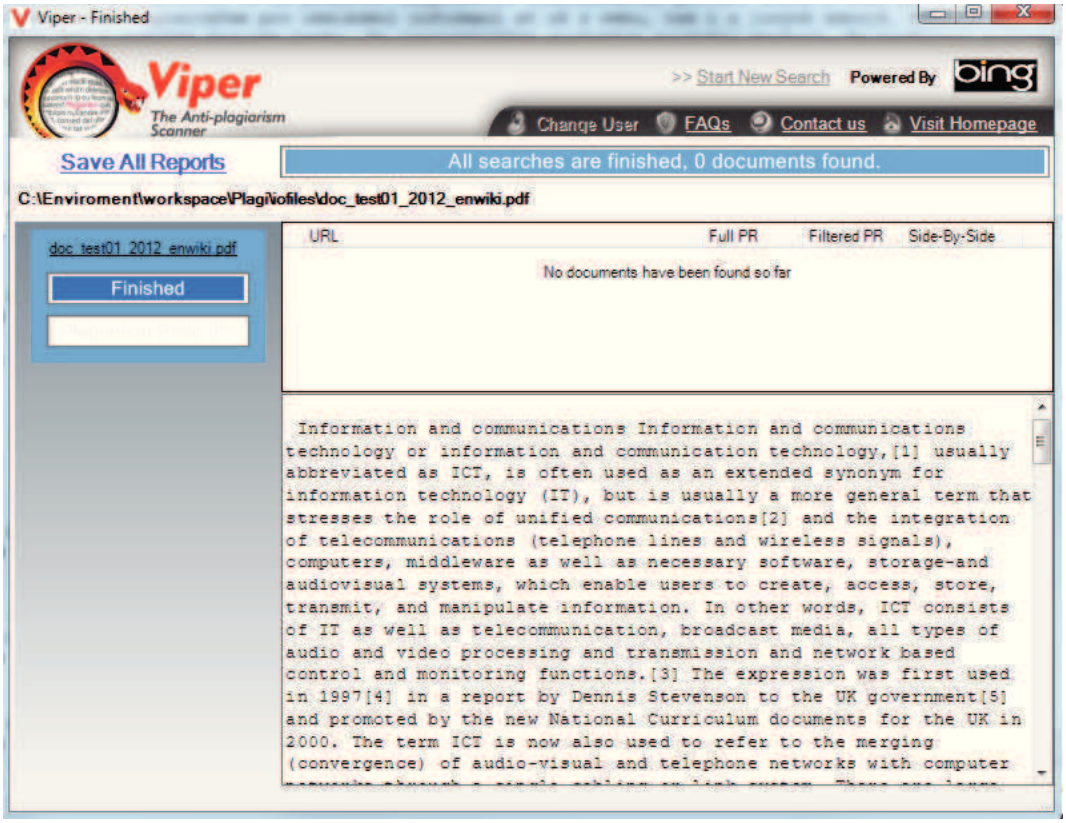
Program Plagiarism-Detector prošel testem nejúspěšněji, protože se mu podařilo nejpřesněji identifikovat původní zdroj a také pracoval nejkratší dobu. Je zjevné, že úspěch byl docílen, především díky tomu, že disponuje kvalitně implementovaným repositářem pro ukáládání informací ať už z webu, tak i z jiných zdrojů. Ukázalo se, že tento princip se v současné době jeví jako nejvhodnější, protože umožňuje prohledat a porovnat obrovské

Program	**Dobra zpracování [minuty]**	**Počet nalezených výsledků**	**Přesnost [%]**
Viper	6:05	0	0
PlagiWeb Tool(anglický dokument)	15:28	12	20
PlagiWeb Tool(český dokument)	12:14	9	55
Plagiarism-Detector	2:13	18	85(99)

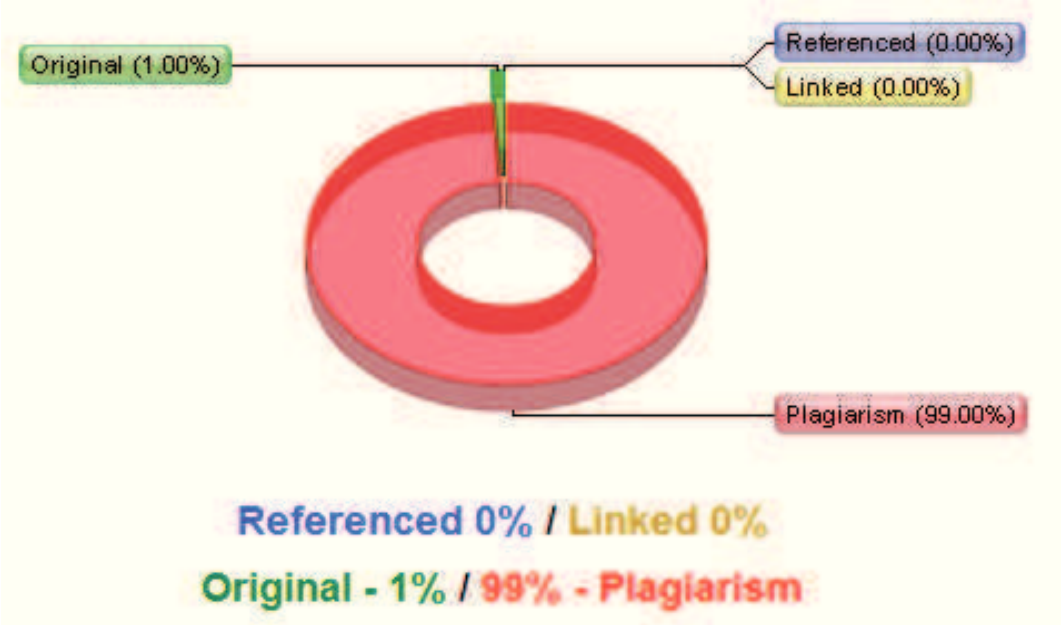
Tabulka 12: Porovnání programů

množství dat v krátkem časovém úseku. Dokonce se mu podařilo najít dokument na úložišti *www.scribd.com*, který odpovídal použitému textu z učebnice sociologie, ovšem byl zde uveřejněn pouze článek a ne konkrétní učebnice, takže podle mého názoru v tomto případě nešlo o plagiát. To samozřejmě souvisí s tím, že každá aplikace má informativnní charakter a je pouze na každém posuzujícím člověku jak se získanými informacemi naloží. Na internetových stránkách produktu zmiňují, že využívají n-gramového repozitáře, ale podrobnosti o jeho implementaci už nezveřejňují (jedná se o komerční subjekt, který nemá potřebu zpřístupňovat své know-how). Program je možné stáhnout ze internetových stránek http://plagiarism-detector.com/.

Tabulka 12 uvádí výsledky testů porovnávaných aplikací. Pro PlagiWeb Tool, jsou zde uvedeny dva výsledky, které rozlišují zpracování česky psaného a anglicky psaného dokumentu. Hodnota „*99*“ u Plagiarism-Detectoru znamená, jak byl dokument ohodnocen samotnou aplikací, ale po důkladném prozkoumání uvedených zdrojů, byl výsledek přepočítán.



Obrázek 20: Výsledky programu Viper



Obrázek 21: Plagiarism-Detector zobrazené výsledky

```
url: www.experiencefestival.com/a/Engine/id/1894993 podobnost: 17,29
url: hartoyo.wordpress.com/author/hartoyo podobnost: 2,88
url: hartoyo.wordpress.com/2009/07/07/ict-in-the-learning-of-efl podobnost: 2,88
url: nelvinandrade.blogspot.com podobnost: 2,88
url: en.wikipedia.org/wiki/Engine podobnost: 84,58
url: forums.corvetteforum.com/c3-general/2916269-motor-oil-2.html podobnost: 2,88
url: uk.answers.yahoo.com/question/index?qid=20101110031125AATTHSA podobnost: 17,29
url: uk.answers.yahoo.com/question/index?qid=20101104102240AA3JAhC podobnost: 17,29
url: answers.yahoo.com/question/index?qid=20080513161648AAZujkS podobnost: 31,70
url: answers.yahoo.com/question/index?qid=20101110031125AATTHSA podobnost: 17,29
url: en.wikipedia.org/wiki/Communications_technology podobnost: 11,53
url: answers.yahoo.com/question/index?qid=20101104102240AA3JAhC podobnost: 17,29
url: www.experiencefestival.com/a/Engine_-_History_of_engines/id/606063 podobnost: 8,65
url: www.wiezoekje.com/engin podobnost: 2,88
url: en.wikipedia.org/wiki/Sociology podobnost: 17,29
url: primapump.wordpress.com podobnost: 14,41
url: www.scribd.com/tariqghayyur2/d/49534460-Information-amp... podobnost: 5,76
url: pediaview.com/openpedia/Information_communication_technology podobnost: 11,53
File:                       c:\Enviroment\workspace\Plagi\iofiles\doc_test01_2012_enwiki.pdf
Author:                     test01
Year:                       2012
Count KeyWords:             422
Count of Founded Links:     131
Count of Compared Pages:    78
Count Winnowed:             347
Count Hashes:               1387
Count of Detected Links:    18
```

Obrázek 22: Výsledek z programu PlagiWeb Tool

6.3.5 Rozbor výsledků z PlagiWeb Tool

Následuje vysvětlení výstupu z programu PlagiWeb Tool zobrazeného na obrázku 22. Výpis je tvořen ze dvou částí, kde v první je seznam nalezených webových stránek, na kterých byla nalezena nějaká shoda a na stejném řádku za každou adresou následuje procentuální vyjádření shody. Pod sekcí s výpisem webových adress, následují informace o zpracovaném souboru:

- File - název zpracovávaného souboru,
- Author - login autora dokumentu (vyextrahován z názvu souboru),
- Year - rok absolvování (také vyextrahován z názvu souboru),
- Count KeyWords - počet nalezených klíčových slov,
- Count of Founded Links - počet nalezených odkazů vyhledávačem,
- Count of Compared Pages - počet kontrolovaných webových stránek (jsou redukovány duplicitní webové odkazy a odkazy, které nelze otevřít např. nepodporované typy dokumentů),
- Count Winnowed - počet n-gramů, které slouží pro porovnávání (redukovány z celkového počtu pomocí lokálního winnowingu),
- Count Hashes - celkový počet n-gramů v dokumentu
- Count of Detected Links - počet odkazů, na kterých byla nalezena shoda.

Doba zpracování jednoho dokumentu, je závisla na jeho celkovém obsahu a počtu porovnávaných webových stránek. V případě textu, který obsahuje 1000 slov a je porovnáván se 100 webovými stránkami o přibližně stejné délce jako je porovnávaný dokument, trvá doba zpracování cca 5 - 7 minut. Při porovnávání textů s větším objemem a s více webovými stránkami můžeme dosáhnout doby zpracování 30 minut a více. Nejslabším článkem, který má přímý vliv na dobu běhu je dotazování se vyhledávací služby, která má časové omezení 1 dotaz za 1,5 vteřiny a převod slov do jejich kořenového tvaru využívající externí databázi.

7 ZÁVĚR

Hlavním cílem práce bylo vytvořit pomůcku pro středoškolské pedagogické pracovníky pro odhalování plagiátorství a zároveň objasnit, proč vůbec k tomuto jevu dochází, jak vzniká a jak je možné proti němu bojovat a předcházet mu. Seznámit čtenáře s existujícími softwarovými aplikacemi sloužících pro jejich odhalování, jejich základními principy a metodami zpracování textu, které se využívají pro analýzu.

V praktické části bylo cílem vytvořit fungující aplikaci, která bude schopna odhalit plagiátorství prostřednictvím sítě Internet s využitím vyhledávací služby Bing. Zároveň ověřit, jestli je vůbec takové vyhledávání možné oproti použití lokalního repozitáře. Aplikace měla podporovat jediný vstupní formát s možností nastavit zpracování jednotlivých částí dle přiloženého modelu.

Podstatnou částí bylo nastudovat vhodné metody potřebné k realizaci celého programu. A na základě výsledků z jednotlivých experimentů v kapitole 6.1 a 6.2 vybrat tu nejvhodnější. Následně provést porovnání s několika vybranými produkty uvedenými v kapitole 2.4.1 a vyhodnotit jestli byl výběr jednotlivých technik vhodný. Aplikace sice obstála v porovnání s produktem Viper, který nenalezl žádne shody, ale v porovnání s Plagiarism-Detectorem, nedosahovala tak excelentních výsledků. Výsledkem tedy je, že lze využít internetu jako zdroje, který obsahuje dokumenty s různými informacemi, ale pro jejich efektivní a přesné porovnávání je třeba využít sofistikovanějších metod pro analýzu textu (např. singulární rozklad ve spojení s latentní sémantickou analýzou) ve spojení s vhodným repozitářem (např. s využitím n-gramů). Všechny uvedené cíle byly splněny a vznikl systém PlagiWeb Tool, který je schopen odhalovat podobné dokumenty prostřednictvím sítě Internet, díky kterému mohou, ale nemusí být posuzované dokumenty označeny za plagiát. Zároveň jsem ověřil že je možné využít tak nehomogenního repozitáře dokumentů s použitím vhodných metod pro detekci klíčových slov, zpracování textu, které byly testovány a uvedeny v kapitole 6 ve spojení s webovou službou.

Aplikace v současné době disponuje RAKE metodou detekce klíčových slov a zpracováním textu v podobě lokálního winnowingu. Ovšem není problém systém rozšířit o další metody a podporu více typů zpracovávaných dokumentů a zvýšit tak jeho možnosti. Samozřejme aplikace nedosahuje tak velkých kvalit jako systém Theses.cz, který je stále vyvíjen a rozšiřován o další prvky skupinou lidí, kteří se dohromady zabývají všemi jeho možnostmi. Důvodem proč vznikl tento systém, který by se mohl jevit jako duplicitní (existuje totiž mnoho podobných a lepších systémů, jak už bylo otestováno), je jeho nízká cena v porovnání s uvedenými a porovnanými programy a také fakt, že ani jeden z nich nepodporuje český jazyk a neumí jej zpracovat. Nicméně i tato menší aplikace v porovnání s ostatními systémy má potenciál stát se více flexibilní, výkonnější a použitelnou. A tím tak může ještě více zlepšit své šance na poli prevence a v boji proti plagiátorství.

8 Reference

[1] BERRY, Michael J. *Text mining: applications and theory.* 1. vyd Oxford: Wiley, c2010, 207 s. ISBN 978-0-470-74982-1.

[2] WEISS, Sholom M. *Text mining: predictive methods for analyzing unstructured information.* New York: Springer, 2005, 237 s. ISBN 03-879-5433-3.

[3] FELDMAN, Ronen. *The text mining handbook: advanced approaches in analyzing data.* New York: Cambridge University Press, 2007, 410 s. ISBN 05-218-3657-3.

[4] CHÝLA, Roman. Detekce plagiátorství.*Ikaros Elektronický Časopis o Informačni Společnosti / Ústav Informačních Studií a Knihovnictví Praha.* 2009, roč. 13, č. 2. ISSN 1212-5075. Dostupné z: http://www.ikaros.cz/node/5253

[5] CHÝLA, Roman. Detekce plagiátorství.*Ikaros Elektronický Časopis o Informačni Společnosti / Ústav Informačních Studií a Knihovnictví Praha.* 2009, roč. 13, č. 3. ISSN 1212-5075. Dostupné z: http://www.ikaros.cz/node/5308

[6] MANNING, Christopher D. *Foundations of statistical natural language processing.* Cambridge: MIT Press, c1999, 680 s. ISBN 02-621-3360-1.

[7] BAEZA-YATES, R a Berthier de Araújo Neto RIBEIRO. *Modern information retrieval.* New York: ACM Press, 1999, 513 s. ISBN 02-013-9829-X.

[8] BRANDEJS, Michal, Jitka BRANDEJSOVÁ, Růžena KRHUTKOVÁ, Zuzana MIKULÁŠOVÁ a Lucie PEKÁRKOVÁ. Kontrola plagiátů v seminárních pracích prostřednictvím Odevzdej.cz.*Ikaros Elektronický Časopis o Informačni Společnosti / Ústav Informačních Studií a Knihovnictví Praha.* 2009, roč. 13, č. 8. ISSN 1212-5075. Dostupné z: http://www.ikaros.cz/node/5641

[9] KŘIPAČ, Miroslav, Michal BRANDEJS, Jan KASZPARK a Jitka BRANDEJSOVÁ. *Zpravodaj ÚVT MU Bulletin pro zájemce o výpočetní techniku na Masarykově univerzitě: Systém pro odhalování plagiátů na českých vysokých školách.* Brno: Ústav výpočetní techniky MU, 5. červen 2008, XVIII, č. 5. ISSN 1212-0901. Dostupné z: http://www.ics.muni.cz/bulletin/articles/585.html

[10] DEITEL, Paul J a Harvey M DEITEL. *Java: how to program.* 8th ed. Upper Saddle River, N.J.: Pearson Prentice Hall, c2010, 1506 s. ISBN 01-360-5306-8.

[11] SIERRA, Kathy a Bert BATES. *SCJP Sun certified programmer for Java 6 study guide: exam (310-065).* New York: McGraw-Hill, c2008, 851 s. ISBN 00-715-9106-0.

[12] KRUCHTEN, Philippe. *The rational unified process: an introduction.* 3rd ed. Upper Saddle River: Addison-Wesley, 2004, 310 s. ISBN 03-211-9770-4.

[13] AGRESTI, Alan. *Categorical data analysis.* 2nd ed. Hoboken: Wiley, 2002, 710 s. Wiley series in probability and statistics. ISBN 04-713-6093-7.

[14] GUSFIELD, Dan. *Algorithms on strings, trees, and sequences: computer science and computational biology.* New York: Cambridge University Press, 1997, 534 s. ISBN 05-215-8519-8.

[15] Zákon o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon). *Národní knihovna České republiky: Autorské právo a knihovny* [online]. 1. prosince 2000 [cit. 2012-04-21].
Dostupné z: http://www.nkp.cz/o_knihovnach/AutZak/Index.htm

[16] Úvod do problematiky plagiátorství. *Infogram portál pro podporu informační gramotnosti* [online]. 2008 [cit. 2012-04-21].
Dostupné z: http://www.infogram.cz/findInSection.do?sectionId=1115&categoryId=1161

[17] *Recognizing and Avoiding Plagiarism* [online]. 2005 [cit. 2012-04-28].
Dostupné z: http://plagiarism.arts.cornell.edu/tutorial/index.cfm

[18] STROSSA, Petr. *Vybrané kapitoly z počítačového zpracování přirozeného jazyka.* Vyd. 1. Opava: Slezská Univerzita v Opavě, 1999, 277 s. ISBN 80-724-8041-3.

[19] ČEŠKA, Zdeněk. *Využití techniky náhodného indexování v oblasti detekce plagitátů.* [online]. 2010 [cit. 2012-05-03].
Dostupné z: http://textmining.zcu.cz/publications/NahodneIndexovani-Plagiaty-ITAT2009.pdf

[20] Řešení problematiky v České republice. *Infogram portál pro podporu informační gramotnosti* [online]. 2008 [cit. 2012-04-28].
Dostupné z: http://www.infogram.cz/findInSection.do?sectionId=1115&categoryId=1176

[21] *Vysokoškolské kvalifikační práce* [online]. 2011 [cit. 2012-04-22].
Dostupné z: http://www.theses.cz/

[22] Plagiarism Today. *The 3 Uses for Plagiarism Detection Tools* [online]. 1. března 2011 [cit. 2012-04-22].
Dostupné z: http://www.plagiarismtoday.com/2011/03/03/the-3-uses-for-plagiarism-detection-tools/

[23] TÉMA: Národní registr VŠKP a systém na odhalování plagiátů. BRANDEJSOVÁ, JITKA. *Čtenář - měsíčník pro knihovny* [online]. leden 2009 [cit. 2012-04-22].
Dostupné z: http://ctenar.svkkl.cz/clanky/2009-roc-61/01-2009/tema-narodni-registr-vskp-a-system-naodhalovani-plagiatu-51-311.htm

[24] BIALAS, Ondřej. *Nástroj pro identifikaci plagiátů a podobných dokumentů* [online]. Ostrava, 2011 [cit. 2012-04-30].
Dostupné z: http://dspace.vsb.cz/handle/10084/87089. Diplomová práce. Vysoká škola báňská - Technická univerzita Ostrava.

[25] TÝN, Pavel. *Návrh repositáře studijních projektů se zaměřením na rozpoznávání podobných prací* [online]. Ostrava, 2009 [cit. 2012-04-30]. Dostupné z: http://dspace.vsb.cz/handle/10084/75598. Bakalářská práce. Vysoká škola báňská - Technická univerzita Ostrava.

[26] ZDRAŽILOVÁ, Iva. *Problém plagiátorství na vysokých školách* [online]. 4. května 2008 [cit. 2012-04-25].
Dostupné z: http://www.inflow.cz/problem-plagiatorstvi-na-vysokych-skolach

[27] KLIMÁNEK, Oldřich. *Silnější vyhledávání slov pomocí nové matematické techniky* [online]. 10. duben 2009 [cit. 2012-05-03].
Dostupné z: http://www.scinet.cz/silnejsi-vyhledavani-slov-pomoci-nove-matematicke-techniky.html

[28] *Vynálezce hromosvodu* [online]. 19. června 2009 [cit. 2012-04-29].
Dostupné z: http://www.ptejteseknihovny.cz/uloziste/uog001/vynalezce-hromosvodu

[29] *Hans Lippershey z Middelburgu, skutečný vynálezce dalekohledu* [online]. 14. března 2012 [cit. 2012-04-29].
Dostupné z: http://www.enviweb.cz/rss/priroda/53864/hans-lippershey-z-middelburgu-skutecny-vynalezce-dalekohledu

[30] *Bing API* [online]. 2012 [cit. 2012-04-29].
Dostupné z: http://msdn.microsoft.com/en-us/library/dd900818.aspx

[31] *Instalace Postgre SQL.* [online]. 12. březen 2012 [cit. 2012-04-27].
Dostupné z: http://postgres.cz/wiki/Instalace_PostgreSQL

[32] Installing MySQL on Microsoft Windows. *MySQL the worldś most popular open source database* [online]. 2012 [cit. 2012-04-27].
Dostupné z: http://dev.mysql.com/doc/refman/5.5/en/windows-installation.html

[33] *How do I install Java ?* [online]. 2012 [cit. 2012-04-28].
Dostupné z: http://www.java.com/en/download/help/download_options.xml